



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## DETEKCE DEFEKTŮ V DATECH KAMERY

DETECTION OF DEFECTS IN CAMERA DATA

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Svätopluk Blažej

### VEDOUcí PRÁCE

SUPERVISOR

Ing. Ilona Janáková, Ph.D.

BRNO 2019

# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**  
Ústav automatizace a měřicí techniky

**Student:** Svätopluk Blažej

**ID:** 162128

**Ročník:** 3

**Akademický rok:** 2018/19

**NÁZEV TÉMATU:**

## Detekce defektů v datech kamery

### POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit robustní systém, který bude v obrazech z kamery detekovat nově vzniklé i existující defekty. Předpokládána je detekce minimálně dvou typů z těchto defektů: rozostření, neprůhledné (bahno, vlákna,...) i průsvitné (voda, sůl, olej,...) nečistoty, mechanické poškození optické soustavy (praskliny, škrábance,...), vady senzoru atd. Výstupem by měla být klasifikace, zda daný snímek obsahuje defekt či ne, odhad velikosti defektu (např. % z plochy) a lokalizace defektu v obraze.

1. Seznamte se s danou problematikou.
2. Proveďte rešerši metod zpracování obrazu vhodných pro tuto úlohu.
3. Poříděte dostatečně rozsáhlou a pestrou databázi snímků/sekvencí.
4. Vyberte vhodné metody, případně navrhnete vlastní metody detekce jednotlivých defektů.
5. Metody implementujte (nejlépe s využitím knihovny OpenCV).
6. Vše otestujte, stanovte omezující podmínky, zhodnoťte.

### DOPORUČENÁ LITERATURA:

REEVES, Stanley J. Image Restoration: Fundamentals of Image Restoration. Academic Press Library in Signal Processing: Vol. 4 [online]. Elsevier, 2014, 2014, s. 165-192 [cit. 2018-09-14]. Academic Press Library in Signal Processing. ISBN 9780123965011. Dostupné z: <https://www.sciencedirect.com/science/article/pii/B9780123965011000066>

CHONG, R. M., TANAKA, T.: Image Extrema Analysis and Blur Detection with Identification, Signal Image Technology and Internet Based Systems 2008. SITIS '08. IEEE International Conference on, pp. 320-326, 2008.

**Termín zadání:** 4.2.2019

**Termín odevzdání:** 20.5.2019

**Vedoucí práce:** Ing. Ilona Janáková, Ph.D.

**Konzultant:**

**doc. Ing. Václav Jirsík, CSc.**  
předseda oborové rady

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Cieľom bakalárskej práce je implementácia algoritmov schopných rozpoznať rôzne typy defektov kamery za účelom analýzy kvality snímaného obrazu. Bakalárska práca zároveň vyhodnocuje rýchlosť a presnosť navrhnutého algoritmu na datasete obrázkov rôznych defektov.

## **KLÍČOVÁ SLOVA**

Kamerový systém, defekt, scéna, obraz.

## **ABSTRACT**

The aim of the bachelor thesis is to implement algorithms able to recognize different types of camera defects in order to analyze the quality of the scanned image. At the same time, the bachelor thesis evaluates the speed and accuracy of the proposed algorithm on the dataset of images of various defects.

## **KEYWORDS**

Camera system, defect, accuracy, camera image

## PROHLÁŠENÍ

Prohlašuji, že svoji bakalářskou práci na téma Detekce defektů v datech kamery jsem vypracoval samostatně pod vedením vedoucí bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Ďakujem vedúcej bakalárskej práce Ing. Ilone Janákovéj Ph.D. za odbornú pomoc, konzultácie a rady k tvorbe. Zároveň by som chcel poďakovať pánovi Ing. Martinovi Šimonovi za odborný dozor a pomoc pri zbere dát.

# OBSAH

<b>1. Úvod</b>	<b>1</b>
<b>2. Spôsoby detekcie defektov kamery</b>	<b>2</b>
2.1. Rozdelenie a detekcia defektov statických systémov .....	2
2.1.1. Analýza defektu na základe matematického popisu .....	3
2.1.2. Analýza defektu na základe použitia neurónových sietí.....	6
2.1.3. Principal component analysis .....	11
2.1.4. Normalized cross-correlation.....	13
2.2. Rozdelenie a detekcia defektov dynamických systémov.....	15
2.2.1. Optický tok .....	15
<b>3. Praktická implementácia metód detekcie defektu</b>	<b>18</b>
3.1. Popis algoritmu .....	18
3.1.1. Vstup algoritmu .....	23
3.1.2. Výstup algoritmu .....	25
<b>4. Vyhodnotenie navrhnutého algoritmu</b>	<b>28</b>
4.1. Intersection over Union (IoU).....	28
4.2. Vyhodnotenie priemernej rýchlosti algoritmu .....	30
4.3. Vyhodnotenie detekcie algoritmu .....	30
4.3.1. Vyhodnotenie navrhnutých metód pri referenčnom zázname .....	30
4.3.2. Detekcia priesvitných nečistôt .....	32
4.3.3. Detekcia nepriesvitných nečistôt .....	36
<b>5. Záver</b>	<b>40</b>

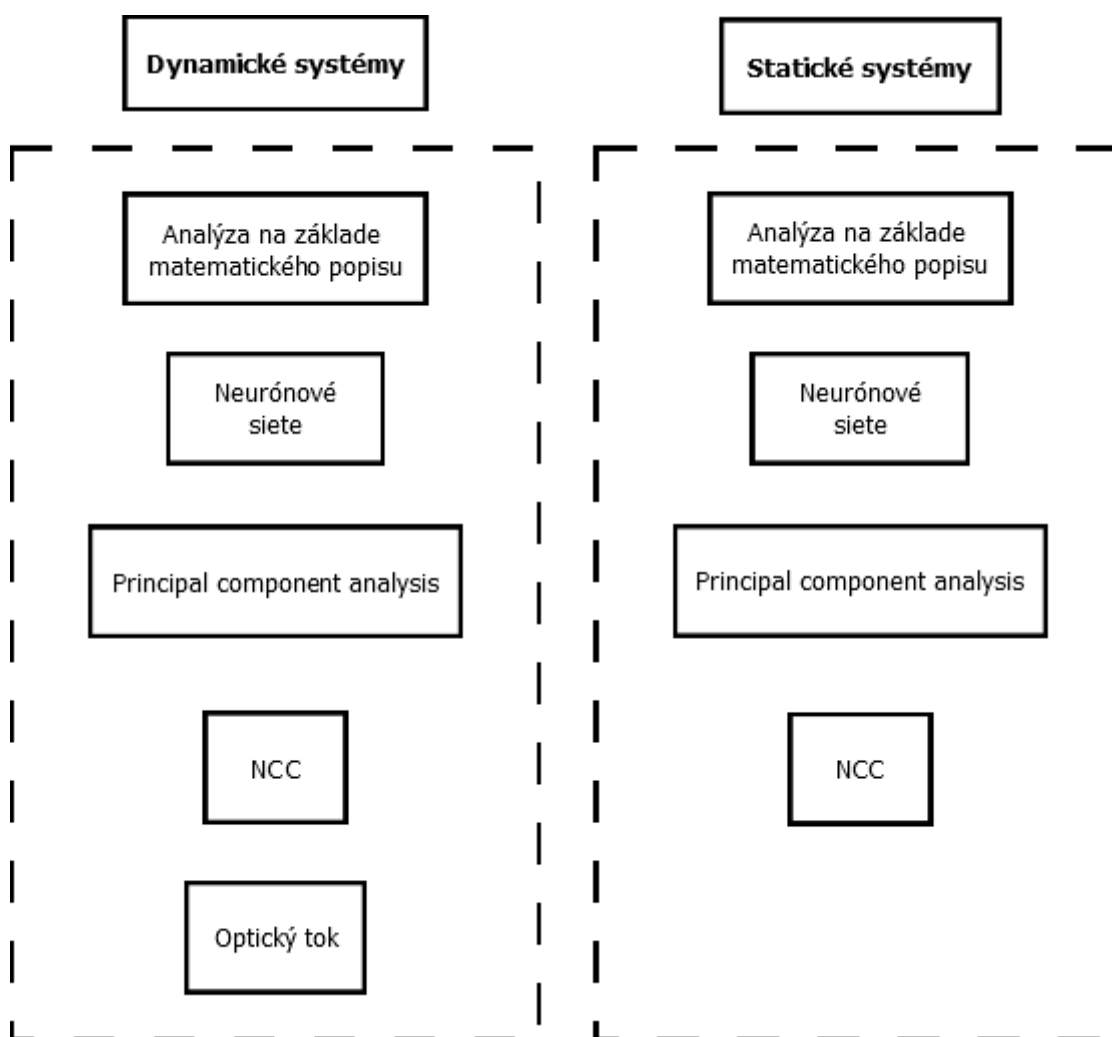


# 1. ÚVOD

V dnešnej dobe sa stávajú kamerové systémy nenahraditeľnou súčasťou každodenného života. Kamery nahrádzajú ľudí vo výrobných linkách, snímajú náš pohyb na uliciach prípadne nám pomáhajú bezpečne riadiť vozidlo alebo dohliadajú na náš majetok. Pri každom prípade použitia je dôležitá spoľahlivosť kamery. Používateľ musí mať istotu v tom, že pokiaľ je kamerový záznam hlavný zdroj informácií o tom čo sa práve deje, tak jeho pravdivostná hodnota bude maximálna. Veľkým problémom pri kamerových systémoch je zašpinenie, poškrabanie prípadne iné znehodnotenie kamerového aparátu. Vzniknuté artefakty môžu spôsobiť skreslenie a mylný výstup vyhodnocovacieho algoritmu, čo môže spôsobiť veľké škody. Autonómne vozidlo musí vedieť o tom, či záznam z kamery je v dostatočnej kvalite pre vyhodnocovanie okolia vozidla. Pri analýze scény sa vždy dbá na jeho jednoduchosť a rýchlosť. Algoritmus je súčasťou komplexného primárneho algoritmu, ktorý je určený k obsluhu motorového vozidla. Na základe tohto predpokladu bola snaha algoritmus implementovať v čo najjednoduchšej forme.

## 2. SPÔSOBY DETEKCIE DEFEKTOV KAMERY

Cieľom kapitoly rešerše základných metód detekcie rôznych typov defektov kamery, ktoré sú použiteľné na daný problém. Kapitola popisuje teoretické vysvetlenie podstaty danej metódy a jej uplatnenie na detekciu defektu. Základný popis metód za účelom rozpoznania defektov je zobrazený na obr. 2.1.



Obr. 2.1 Základné metódy určené k rozpoznávaniu defektov

### 2.1. Rozdelenie a detekcia defektov statických systémov

Statický systém je systém v čase invariantní. Čo sa týka analýzy obrazu tak statickým systémom môže byť jedna snímka. Systém neobsahuje informáciu o dynamike defektu. Pokiaľ sa na objektive vyskytuje napríklad vlas prípadne nečistota, v čase vzniku

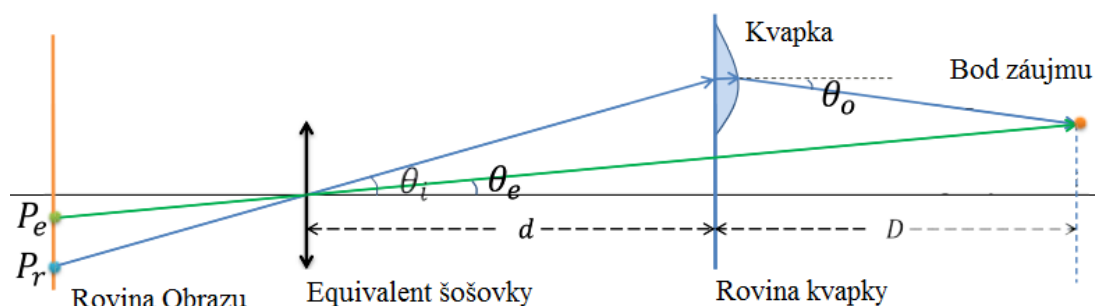
statického záznamu, tak systém nepozná ako sa daný objekt chová v závislosti na prostredí a teda ho môže vyhodnotiť ako defekt kamery. V dynamických systémoch je defekt kamery takmer nemenný v porovnaní k meniacemu sa okoliu.

### 2.1.1. Analýza defektu na základe matematického popisu

Základným predpokladom matematického popisu defektu je možnosť jeho popisu viacerým množstvom veličín: veľkosť, guľatosť, tvar, priehľadnosť atp. Je dôležité nájsť maximálne množstvo popisných veličín pre lepšiu klasifikáciu a samotný matematický popis. Defekty sú nasledovne rozdeľované do tried na základe podobnosti v nami zvolenými parametrami. Pri samotnom rozdeľovaní do tried je potom možné použiť prostriedky matematickej regresie napr. k-means metódu. Pri analýze rôznych typov defektov je dôležité si všimnúť ich rozdielne chovanie. Bahno na šošovke spôsobí tmavý fľak na výstupnom obraze pričom voda ho len rozostří, škrabanec má zvyčajne charakter rovnej čiary no ostatné skôr kruhový tvar. [1]

### Detekcia vodných kvapiek

Vodné kvapky prilhané na skle kamery môžu spôsobiť veľkú degradáciu viditeľnosti scény. V systémoch autonómneho riadenia vozidla je dôležitá kvalita snímaného obrazu. Vodné kvapky sa chovajú ako extra šošovky na sklíčku a teda spôsobujú lom a ohyb vstupného svetelného lúču, na základe ktorého sú oblasti zakryté vodnou kvapkou rozmazané. Fyzikálny princíp prechodu svetla cez kvapku je zobrazený na obr. 2.2. Intenzita prechodu svetla kvapkou sa mení v závislosti na čistote vodného roztoku na sklíčku. Čím viac častíc sa v roztoku nachádza tým menšia intenzita svetla po prechode je zaznamenaná optickým aparátom. Na tvar samotného defektu to veľký vplyv nemá. [1]



Obr. 2.2 Fyzikálny popis lomu svetla po prechode kvapkou [1]







Veľkosť a tvar vodných kvapiek je ovplyvnený rýchlosťou pohybu snímacieho systému, rýchlosťou pohybu padajúcej kvapky a rýchlosťou vetra v prostredí. Základný predpoklad podobnosti tvaru kvapiek spočíva v kapilárnom jave, v medzi molekulových väzbách a v zákone zachovania stavu s najnižšou energiou. Na základe vymenovaných javov má vodná kvapka tendenciu formovať zaguľatený tvar, určitú veľkosť danú charakteristikami prostredia a vlastnosťami šošovky prípadne sklička na ktoré kvapky dopadajú. Základné druhy tvaru kvapiek sú popísané na obr. 2.3. Tvar kvapiek je identifikovaný použitím dvoch matematických rovníc a to rovnica popisujúca hladkosť rov. 2.1 a rovnica popisujúca guľatosť rov. 2.2. Rov. 2.1 popisuje zmenu dotyčnice na hranici pričom rov. 2.2 vyjadruje podiel obsahu tvaru a štvorca svojho obvodu. [1]

$$\mathcal{S}(R) = \oint_{x \in \partial R} |d\theta(x)|$$

Rov. 2.1 Vzťah popisujúci hladkosť objektu [1]

$$\mathcal{O}(R) = \frac{\int_{x \in R} dx dy}{\left(\oint_{x \in \partial R} |dx|\right)^2}$$

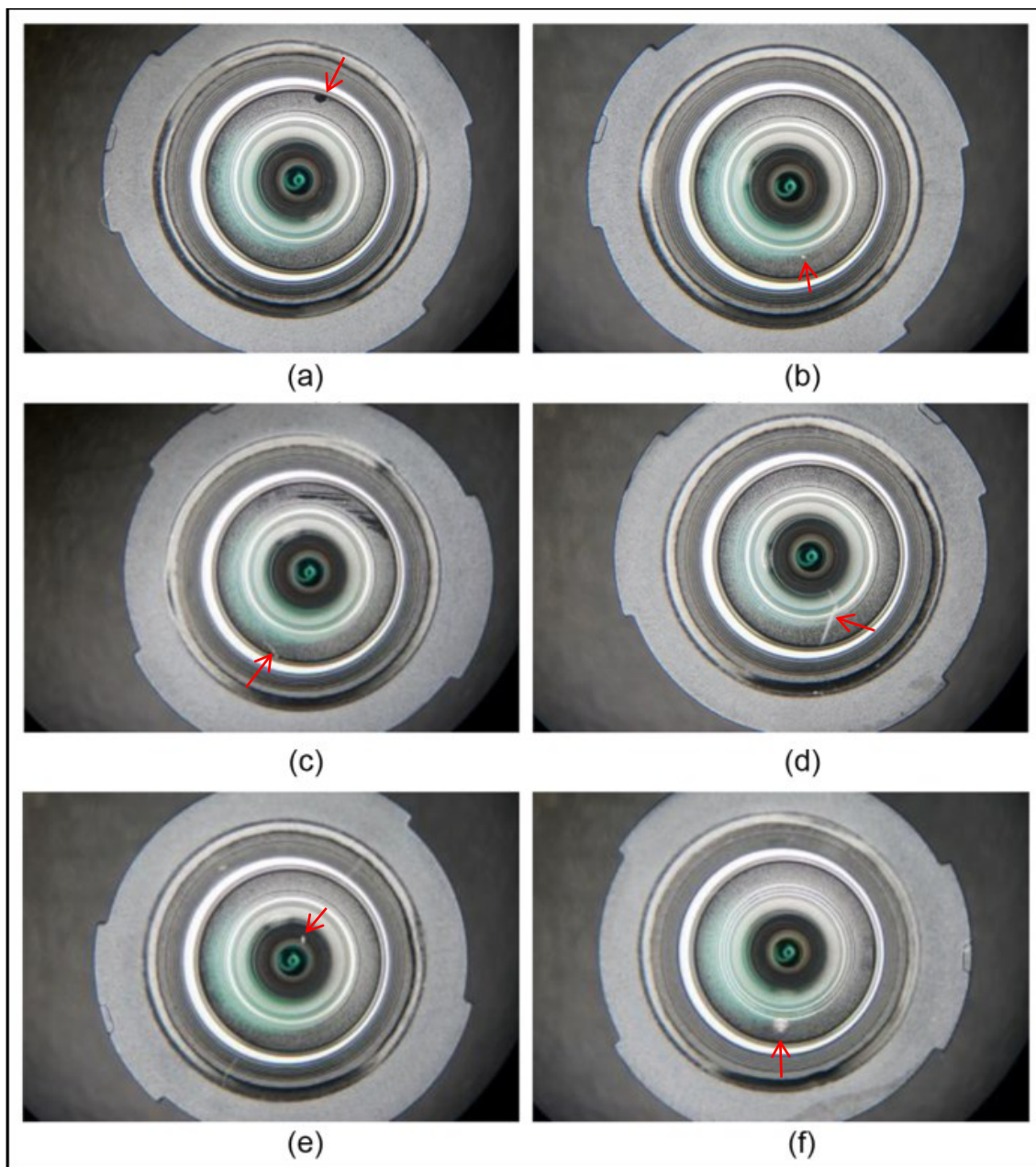
Rov. 2.2 Vzťah popisujúci guľatosť objektu [1]

Tvar						
Hladkosť	$2\pi(6.28)$	$2\pi(6.28)$	$3\pi(9.42)$	11.10	9.41	54.15
Guľatosť	$1/4\pi(0.080)$	0.075	0.050	0.029	0.058	0.016

Obr. 2.3 Matematický popis tvaru kvapky [1]

### Detekcia poškrabaných oblastí a iných nečistôt

Pri niektorých typoch defektov je obzvlášť zložitá ich lokalizácia na scéne. Charakteristiky defektu bývajú príliš podobné charakteristikám scény a preto je nevyhnutné použitie viacerých typov matematických funkcií v po sebe idúcich krokoch určených k lokalizácii defektu, výpočtu jeho charakteristických vlastností a v konečnom prípade jeho klasifikácia do jednotlivých typov. Základné typy defektu sú zobrazené na obr. 2.4. [2]



*Obr. 2.4 Zobrazenie typu defektov: (a) – flak na šošovke; (b), (e), (f) – prachové čiastočky; (c), (d) – poškrabanie šošovky. [2]*

Pri lokalizácii defektov zobrazených na obr. 2.4 sa ako prvá popisná veličina volí intenzita. Je predpoklad, že škrabance prípadne prachové čiastočky zvyšujú intenzitu osvetlenia v danom mieste. Základný popis je založený na tvrdení: intenzita osvetlenia optického čipu v mieste defektu je väčšia ako priemerná intenzita snímanej scény. Zmenu intenzity je možné vypočítať z šedotónového obrázku: porovnávaním aktuálnej hodnoty jas pixelu s priemernou hodnotou jas celého obrázku, adaptívnym porovnávaním, histogramovou analýzou prípadne *OTSU* porovnávaním. Popis intenzity

zlyháva pri veľmi jasných objektoch na scéne prípadne pri nedostatočnom osvetlení. Druhá popisná veličina je zmena kontrastu, a teda detekcia hrán, kde opäť predpokladáme nárast zmeny kontrastu v oblasti defektu v porovnaní s okolím. Obrázok scény sa transformuje na obrázok zmien kontrastu, ktoré sa potom popisujú rôznymi matematickými modelmi. Postup lokalizácie akéhokoľvek defektu je popis jeho chovania a nasledovná klasifikácia vo vstupnom obraze, v našom prípade použitím vlnkovej transformácie a podporných vektorov. [2]

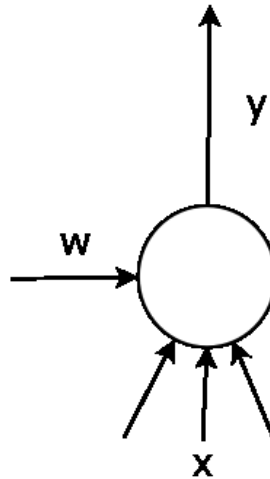
### **2.1.2. Analýza defektu na základe použitia neurónových sietí**

Neurónové siete (NS) patria medzi jedny z robustných algoritmov na detekciu a klasifikáciu rôznych objektov na snímke. S obľubou sa používajú pri objektoch na ktoré je zložitý stanoviť matematický popis ich chovania. Neurónová sieť je matematický model založený na biologickom podklade funkcie neurónov v mozgu.

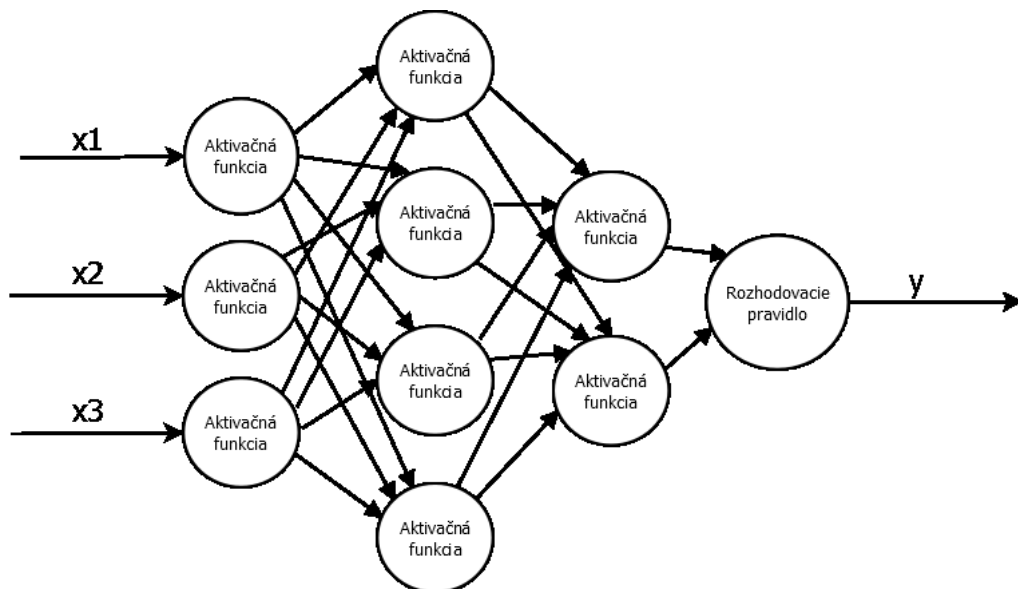
#### **Matematický model NS**

Model neurónu je zobrazený na obr. 2.5. Neurón je základná stavebná jednotka NS, ktorej je priradená určitá funkcia. Do neurónu štandardne vstupuje vážený vstupný signál a hodnota prahu – bias. Každý neurón je prepojený so všetkými neurónmi s nižšej prípadne vyššej vrstvy. Jeho úlohou je prenos signálu na ďalší neurón s určitou váhou, ktorá zvyšuje prípadne znižuje vplyv tohto signálu na nasledujúci neurón. Neuróny s podobnou matematickou funkciou zoskupujeme do vrstiev, typické usporiadanie neurónov je zobrazené na obr. 2.6, pričom každý neurón je popísaný matematickou operáciou, ktorú vykonáva. Medzi matematické operácie patria aj aktivačné funkcie neurónu. Samotný jeden neurón dokáže klasifikovať signál len do dvoch tried, jeho výstupnou hodnotou je logická funkcia. Matematický popis funkcie neurónu je vysvetlený v rov. 2.3, pričom  $u$  je aktivačná funkcia,  $w$  váha a  $x$  vstup. Signál postupuje zo vstupnej vrstvy cez niekoľko medzivrstiev až do finálnej výstupnej vrstvy. Takýmto spôsobom je možná klasifikácia do viacerých skupín. Prvá vrstva sa nazýva vstupnou vrstvou, ktorá vstupuje do ľubovoľného počtu ďalších vrstiev, ktorých úlohou je detekcia príznakov.

Posledná vrstva plní funkciu rozhodovacieho pravidla pre priradenie do jednotlivých tried. [3]



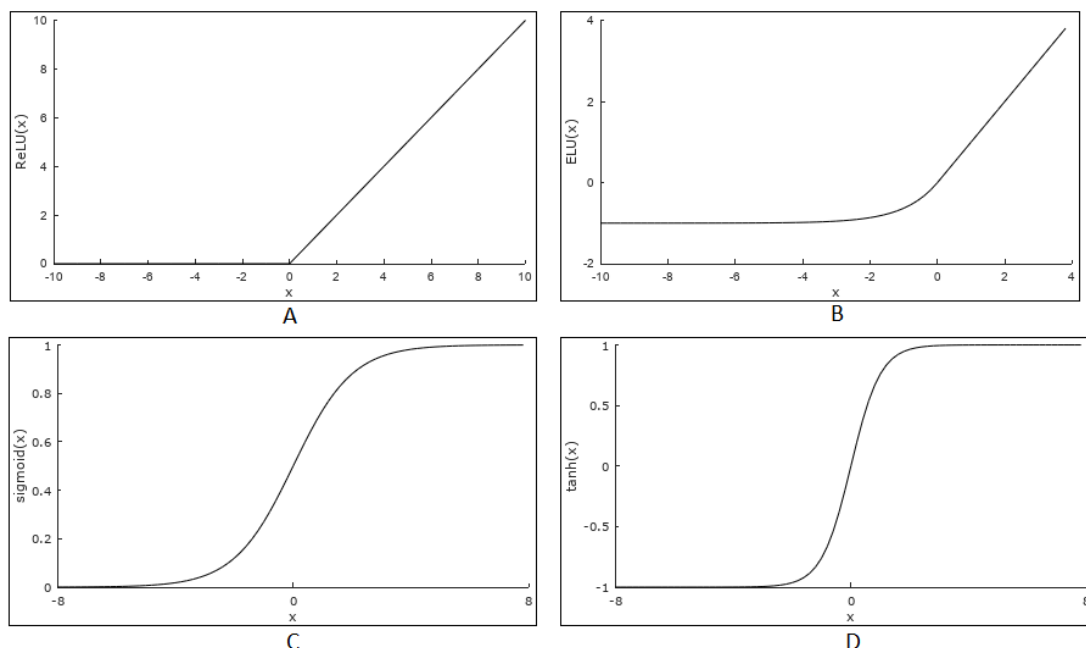
Obr. 2.5 Model neurónu



Obr. 2.6 Príklad modelu neurónovej siete

### Aktivačné funkcie NS

Aktivačné funkcie sa delia do dvoch hlavných kategórií: lineárne alebo nelineárne. Funkcie ktoré neprenášajú záporný vstup ho nahrádzujú nulovou hodnotou. Úlohou aktivačných funkcií je normalizácia ich vstupu a jeho konvertovanie na výstup použitím rôznych pravidiel. Základné typy aktivačných funkcií sú zobrazené na obr. 2.7. [3]



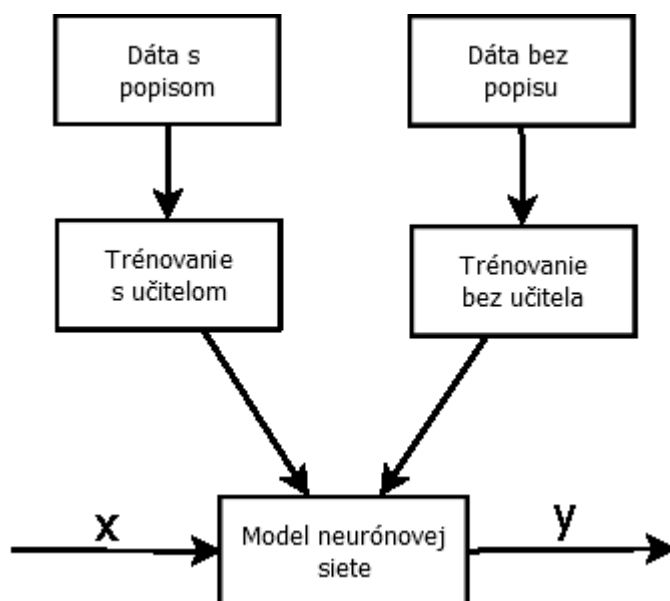
Obr. 2.7 Príklad základných typov aktivačných funkcií [3]

$$y = u \left( \sum_{n=1}^N w_n x_n \right)$$

Rov. 2.3 Vzťah popisujúci funkciu neurónu [3]

Princíp neurónovej siete spočíva v natrénovaní matematického modelu na žiadaný výstup a nasledovná aplikácia tohto modelu na neznámy vstup. Naším cieľom je ju natrénovať tak, aby dávala čo najpresnejšie výsledky. Príkladom je tréning neurónovej siete na rozpoznávanie psa alebo mačky z rôznorodých vstupných snímok. Princíp tréningu a vyhodnocovania neurónových sietí pre tento príklad je zobrazený na obr. 2.8. Je niekoľko modelov neurónových sietí. K základnému rozdeleniu patria NS s učiteľom prípadne bez učiteľa. V NS s učiteľom je definovaný obsah na tréningových dátach. V prípade s mačkou učíme NS na rôznych obrázkoch kde je ale nutné popísať tie obrázky na ktorých sa mačka alebo pes nachádza. Z takýchto obrázkov si NS stanoví, na základe počítania chyby aktuálneho výsledku s požadovaným výsledkom, matematický popisný model. Pri sieťach bez učiteľa, prípady kde nevieme čo hľadáme prípadne je to zložité popísať, si NS sama stanovuje závislosť podobnosti medzi vstupnými dátami a rozdeľuje ich do skupín, takéto učenie vyžaduje dlhší čas a väčší počet obrázkov na naučenie. Ďalším možným rozdelením je rozdelenie podľa matematického popisu klasifikácie na siete konvolučné, priame, priame s rekúziou.

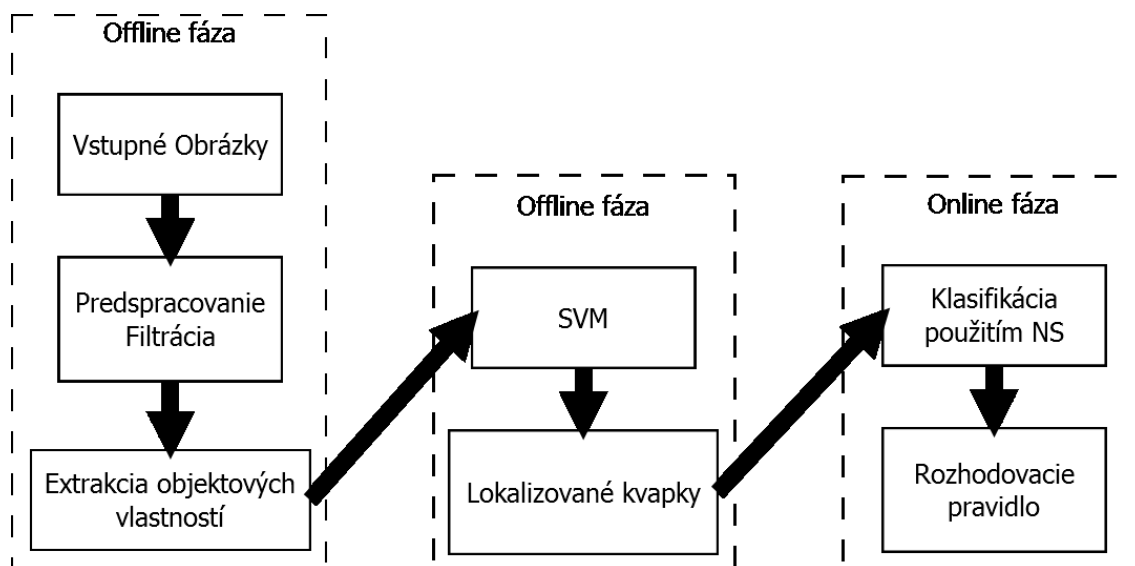




*Obr. 2.8 Vznik a vyhodnocovanie neurónovej siete*

### **Detekcia vodných kvapiek**

Pri tréovaní neurónových sietí s učiteľom je možné využiť dva prístupy: natrénovať sieť na manuálne preddefinované tvary kvapiek prípadne použiť automatickú klasifikačnú a lokalizačnú metódu. Pri učení s učiteľom je nutné na každom datasete obrázkov s vodnými kvapkami vyznačiť miesto každej kvapky. Takto označené dáta je možné potom použiť na natrénovanie neurónovej siete. Vzhľadom na priemerný počet kvapiek na jednom snímku a počet snímok potrebných na natrénovanie siete je táto voľba časovo veľmi náročná. Rýchlejšou alternatívnou metódou k automatickej extrakcii vodných kvapiek z vstupného obrázka je metóda support vector machine (SVM). Algoritmus detekcie kvapiek je zobrazený na obr. 2.9. [4]



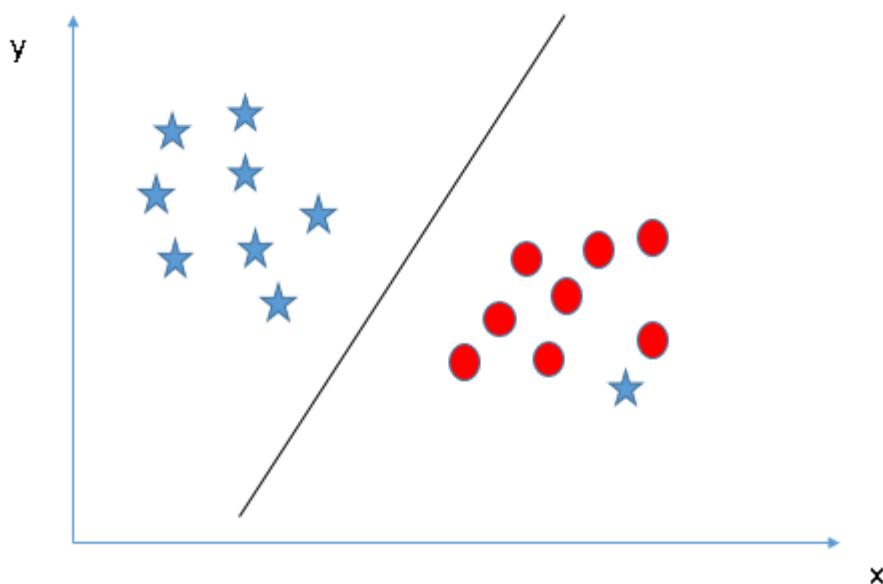
Obr. 2.9 Detekcia kvapiek použitím SVM [4]

Základný princíp algoritmu je automatická lokalizácia väčšiny kvapiek, použitím SVM, na sklíčku, na základe ktorej sa urýchli tvorba modelu NS. Vstupný obrázok sa upraví na čo najväčšiu kvalitu a to zvýraznením kontrastu, filtráciou šumu mediánovým filtrom s malým oknom. Posledným krokom je extrakcia objektových vlastností a teda úprava obrázku za účelom zvýraznenia hrán vodných kvapiek použitím zmeny farebného priestoru na HSV prípadne použitie hranových funkcií. [4]

### Support Vector Machine

Klasifikačná metóda funguje na princípe nájdenia optimálnej úsečky prípadne hyperboly na základe maximalizácie jej vzdialenosti od najbližšieho bodu objektu. Objekty rozdeľuje do  $n$  skupín tak aby vzdialenosť medzi nimi bola čo najväčšia, príklad rozdelenia dát do dvoch skupín použitím SVM je na

obr. 2.10. Vstupné dáta sa delia na trénovacie a testovacie pričom trénovacie dáta sú použité na naučenie klasifikátoru a testovacie na jeho otestovanie. [7]



Obr. 2.10 Separácia dát použitím SVM [7]

### 2.1.3. Principal component analysis

Principal component analysis (PCA) je nástroj umožňujúci transformáciu väčšieho objemu dát na menší pri zachovaní maxima informácie. Vstupné dáta sa na základe korelácie rozdelia do na sebe nezávislých komponentov. Korelácia je viazaná na veľkosť odchýlky korelovaných vstupných hodnôt. Tieto komponenty sa vzájomne porovnávajú na základe ich dôležitosti pričom komponenty s najmenej dôležitou informáciou sa odstraňujú. [6]

#### Princíp Výpočtu

- Rozdelenie vstupných dát na dáta na sebe závislé  $Y$  a dáta na sebe nezávislé  $X$ .
- Normalizácia  $X$  premenných v matici tak aby ich celková stredná hodnota bola nulová a štandardná odchýlka rovná jednej.
- Výpočet vektorov vzájomných odchýlok dát v komponente a odchýlok komponentov medzi sebou podľa rov. 2.4 pričom  $\bar{x}$  je priemerná hodnota a  $N$  celkový počet.
- Odstránenie zbytných množín dát na základe vzájomnej podobnosti.

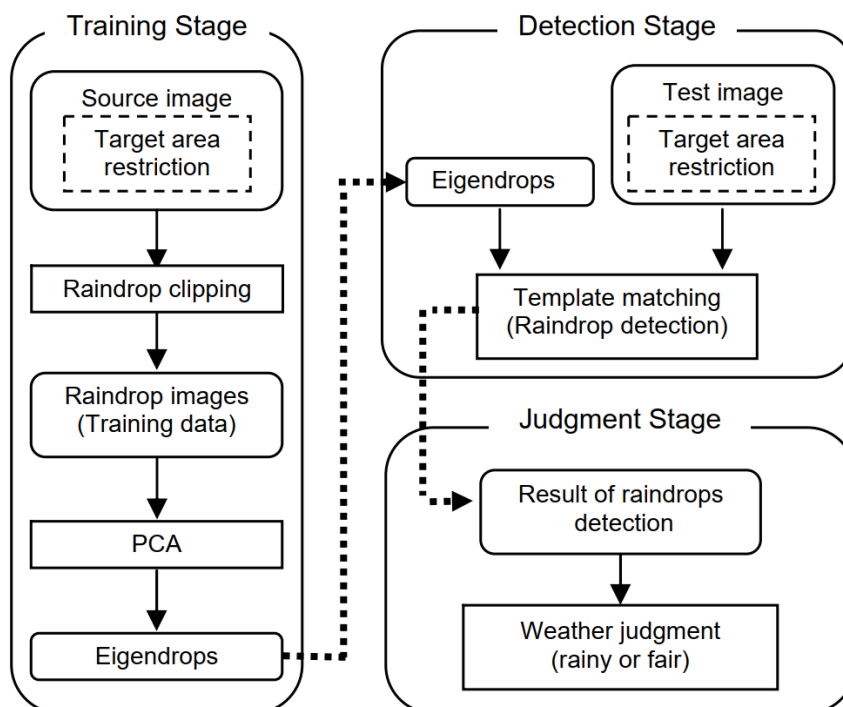
$$var(x) = \frac{\sum (x_i - \bar{x})^2}{N}$$

$$cov(x, y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{N}$$

*Rov. 2.4 Vzťah popisujúci výpočet odchýlok prvkov v komponente v výpočet odchýlky v komponentoch medzi sebou [6]*

### Detekcia vodných kvapiek

Základným výstupom PCA analýzy vodných kvapiek je vygenerovanie reprezentujúcich vodných kvapiek, ktoré sa potom porovnávajú so vstupným obrázkom. PCA analýza vygeneruje popis referenčnej vodnej kvapky. Základnými popisnými parametrami sú: tvar kvapky, popísaný rov. 2.1 a rov. 2.2, rozostrenie obrázku v mieste kvapky, lom a odraz svetla. Objekty z tréningového datasetu snímku porovná medzi sebou na základe vymenovaných vlastností a na základe odchýlky ich rozdelí do patričných komponentov. Komponenty s najväčšou odchýlkou vyfiltruje podľa zadanej prahovej hodnoty a ostatné označí ako vodné kvapky. PCA analýza je len určitým doplnením analýzy podľa matematického popisu. Používa sa ako filter pre odstránenie falošne pozitívnych detekcií a ako nástroj pre rýchle a efektívne stanovenie vlastností referenčných kvapiek. Základný princíp algoritmu je zobrazený na obr. 2.11. [5]



*Obr. 2.11 Popis detekcie vodných kvapiek metódou PCA [5]*

Na obr. 2.11 je zobrazená PCA analýza zložená z troch častí: tréovanie, detekcia a vyhodnotenie. Počas tréovania sú najskôr vstupné obrázky rozdelené na obrázky s kvapkami a bez kvapiek. Z obrázkov sú nasledovne vyrezané len oblasti reprezentujúce vodné kvapky. Tieto obrázky sú potom použité ako tréovací dataset pre PCA analýzu, ktorej výstupom sú referenčné vodné kvapky. Referenčné kvapky sú potom metódou template matching porovnávané s obsahom testovacích obrázkov. Miesta na obrázku zhodné s referenčnými kvapkami sú definitívne označené ako detekované vodné kvapky, na základe ich počtu sa stanovuje kvalita snímaného obrazu. Podobným spôsobom je možné stanoviť akýkoľvek defekt na scéne. Metóda nie je vhodná ako primárna metóda určená k detekcii defektov práve pre nutnosť vyrezania defektov zo snímku v tréovacej fázy. PCA je však vhodné ako doplnkový algoritmus pri lokalizovaných potencionálnych defektoch menej robustnou metódou. V tomto prípade jej použitie napĺňa funkciu dodatočného filtru falošne pozitívnych detekcií. [5]

#### 2.1.4. Normalized cross-correlation

Korelácia signálov vyjadruje ich podobnosť medzi sebou. Výpočet podobnosti dvoch signálov  $x$  a  $y$  je zobrazený na rov. 2.5. Čím viac sú si signály podobné tým väčšie číslo je výstupom rovnice. Nevýhodou daného výpočtu je nutnosť porovnateľnej veľkosti amplitúdy vstupných signálov. Pokiaľ ako vstup korelácie použijeme signály  $x$  a  $\frac{x}{2}$  jej výstupom bude číslo v okolí 20 pričom sa jedná o rovnaký signál. Ďalšou nevýhodou je zložitá interpretácia samotného výstupu korelácie. [8]

$$corr = \sum_{n=0}^{n-1} x[n] * y[n]$$

*Rov. 2.5 Výpočet vzájomnej podobnosti dvoch signálov [8]*

Normalized cross-correlation (NCC) upravuje výstup korelácie na hodnoty v intervale medzi 0 a 1. Výpočet normalizovanej korelácie je znázornený na rov. 2.6.[8]

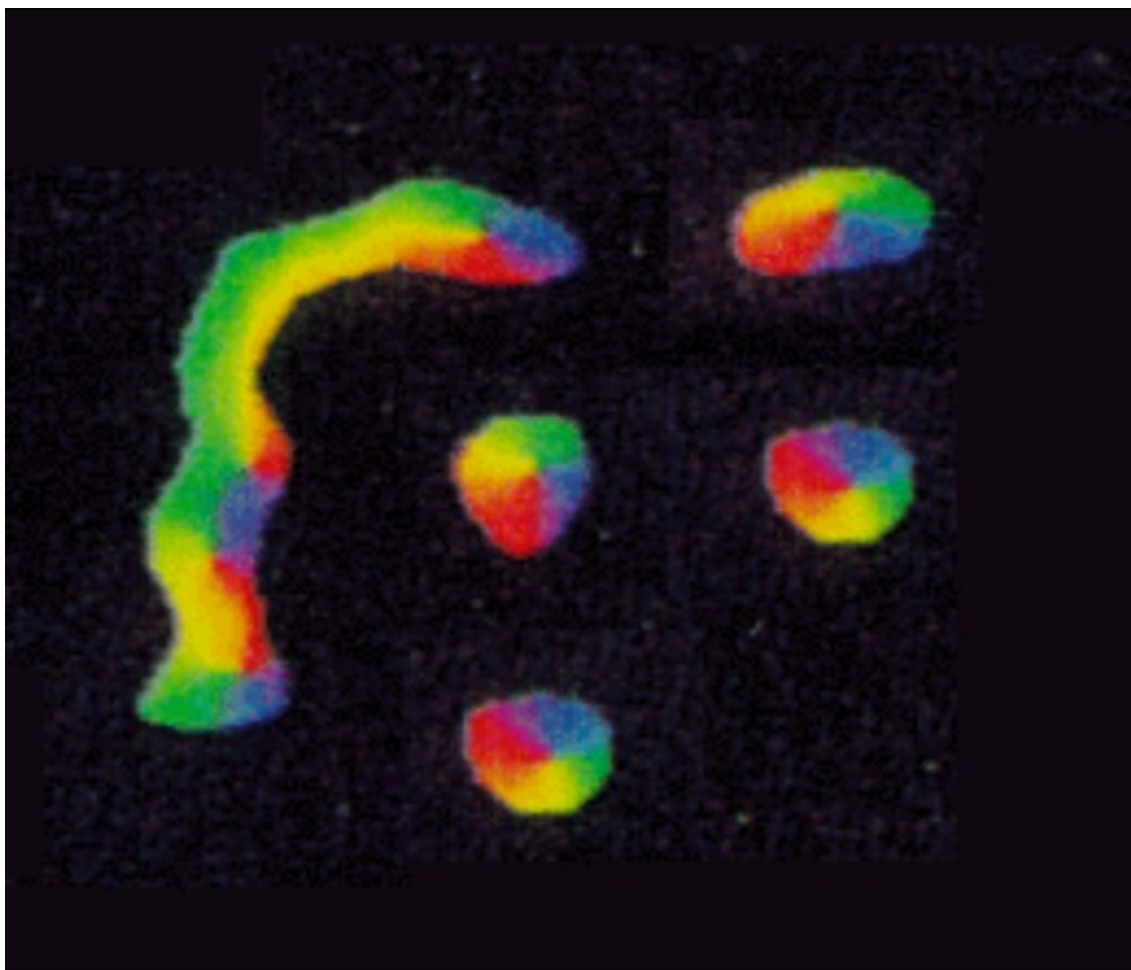
$$NCC = \frac{\sum_{n=0}^{n-1} x[n] * y[n]}{\sqrt{\sum_{n=0}^{n-1} x[n]^2 * \sum_{n=0}^{n-1} y[n]^2}}$$

*Rov. 2.6 Výpočet normalizovanej vzájomnej korelácie dvoch signálov [8]*

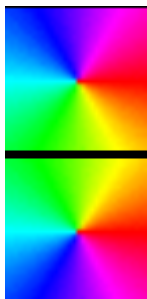
Čím viac sú si signály podobné tým väčšie číslo je výstupom rovnice. Pri úplnej podobnosti signálov je výstup rovný 1. [8]

### Detekcia defektov kamery

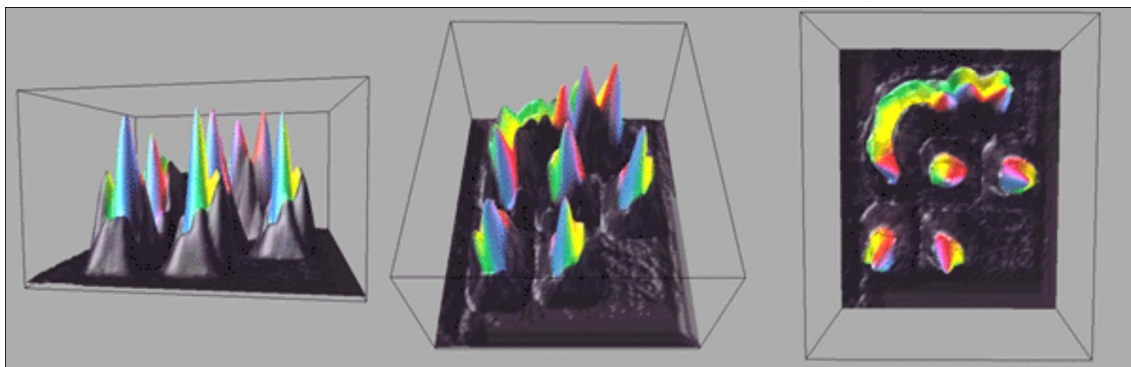
Pri vzájomnej korelácii vstupného obrázku znázorneného na obr. 2.12 s obrázkom vzoru defektu je potrebné vytvoriť plávajúce 2D okno, zobrazené na obr. 2.13, obsahujúce detekovaný vzor. Okno postupne prechádza cez celý vstupný obrázok. V miestach kde je podobnosť vstupného obrázku a okna s defektom vysoká sa výsledok NCC blíži k hodnote 1. Výsledok vzájomnej korelácie je zobrazený na obr. 2.14. [9]



*Obr. 2.12 Vstupný obrázok korelácie [9]*



*Obr. 2.13 Obrázok popisujúci detekovaný vzor [9]*



*Obr. 2.14 Výstupný obrázok hodnôt NCC [9]*

Na výstupnom obrázku obr. 2.14 je možné vidieť vysoké hodnoty korelácie na miestach podobnosti vstupného obrázku a detekovaného vzoru a naopak nízke hodnoty tam kde táto podobnosť nie je.

## **2.2. Rozdelenie a detekcia defektov dynamických systémov**

Medzi dynamický systém patrí akýkoľvek záznam po sebe idúcich zmien scény v závislosti na čase, môže sa jednať o záznam idúceho motorového vozidla, záznam z vonkajších kamier prípadne o jednoduché video z akéhokoľvek zariadenia. Základná podstata spočíva v dynamike snímanej scény a teda v generovaní za sebou idúcej sekvencie obrázkov v ktorej je každý nasledujúci obrázok mierne odlišný od predchádzajúceho. Základný predpoklad detekcie defektov kamery je ich kvazi-statickosť, po určitú dobu snímania záznamu je defekt statický, a teda nemení svoj tvar prípadne miesto na scéne.

Základný princíp metód detekcie v statických systémoch je rovnaký i v dynamických systémoch. Jediný rozdiel je možnosť porovnania za sebou idúcou postupnosť snímok, čo má omnoho väčšiu informačnú hodnotu, a teda odfiltrovanie “dočasných” defektov. V nasledujúcom texte budem popisovať len metódu použiteľnú v dynamických systémoch a teda analýzu optického toku.

### **2.2.1. Optický tok**

Optický tok je približný dvojrozmerný vektor rýchlosti pohybu každého pixelu obrázku v čase. Metóda optického toku umožňuje určiť parametre pohybu, vzdialenosti predmetov v obraze atp. Tento vektor vyjadruje posunutie pixelu z daného snímku na snímok nasledujúci. [10]

Základný predpoklad výpočtu optického toku sa opiera o rovnicu rov. 2.7, ktorá hovorí: pixel šedotónového obrázku o známej intenzite  $I$  v čase  $t$  sa nachádza na mieste definovanom súradnicami  $x$  a  $y$ . v čase  $t+1$  sa ten istý pixel s rovnakou intenzitou posunie o vzdialenosť danú zmenou  $\delta x$  a  $\delta y$ . Praktická ukážka tvrdenia je zobrazená na obr. 2.15, ktorý ukazuje zmenu polohy červeného bodu v čase. [10]

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

*Rov. 2.7 Rovnica pohybu [10]*

Následne po aplikácii Taylorovho rozvoja a matematických úpravách na funkciu intenzity pixelu posunutého v čase dostaneme rov. 2.8., kde  $\mathbf{v}$  reprezentuje vektor optického toku.

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t$$

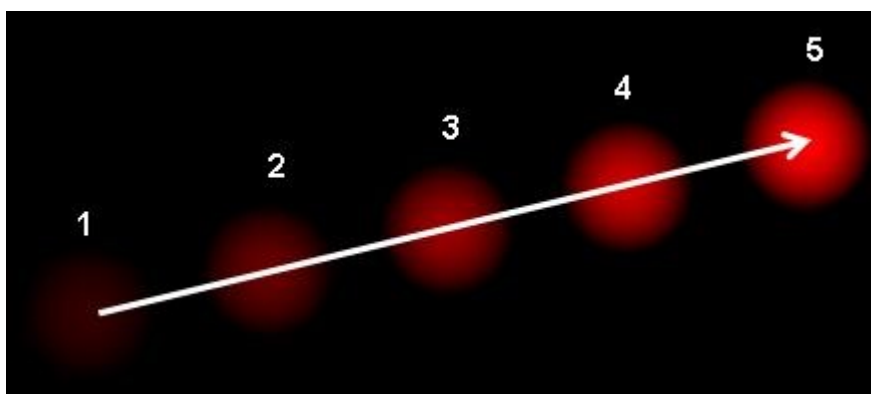
$$\frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t = 0; \delta t = 1$$

$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0$$

$$\left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) * (v_x, v_y) = -I_t$$

$$\nabla I * \mathbf{v} = -I_t$$

*Rov. 2.8 Rovnica optického toku [10]*



*Obr. 2.15 Vektor optického toku[11]*



Základná myšlienka optického toku je detekcia nepohyblivého defektu vzhľadom k pohyblivej scéne.

### 3. PRAKTICKÁ IMPLEMENTÁCIA METÓD DETEKCIE DEFEKTU

Algoritmus popísaný v tejto kapitole slúži k vyhodnocovaniu kvality nasnímanej scény z kamery, ktorá sa nachádza v zadnej časti motorového vozidla. Náš systém je teda dynamického charakteru a je možné použiť metódy určené k detekcii defektu dynamických systémov. Charakter defektov nám umožňuje použiť jednoduchšie prístupy, ktoré sú časovo menej náročné. Algoritmus bude vyhodnocovať kvalitu obrazu v real-time na pozadí primárneho algoritmu.

Navrhovaný algoritmus vychádza z dvoch základných vlastností defektu a to kvazistatickosť a neprítomnosť hrán v oblasti defektu. Kvazistatickosť znamená stálosť prípadne minimálnu zmenu polohy defektu vzhľadom na rýchlosť pohybu vozidla, pričom pri neprítomnosti hrán predpokladám, že po prechode svetelného lúča cez oblasť defektu dôjde k jeho rozmazaniu prípadne úplnej blokácii prechádzajúcich lúčov.

#### 3.1. Popis algoritmu

Algoritmus je naprogramovaný v jazyku Python 2.7 kvôli jednoduchosti a rýchlosti prototypovania. Ako vývojové prostredie bol použitý Linux PyCharm, každopádne algoritmus je multiplatformný. Využíva opensource knižnice určené k práci s obrazom prípadne k analýze dát a to konkrétne OpenCV<sup>1</sup> a NumPy<sup>2</sup>, ďalej pracuje s knižnicou logging pre vytváranie logovacích súborov. Vývojový diagram programu sa nachádza na obr. 3.3. Prvým krokom, je získanie informácií o vstupnom datasete, a to počet vstupných obrázkov, ich názov a rozlíšenie vstupu. Parametre programu sú *-image*, ktorý špecifikuje cestu k obrázkom datasetu, parameter *-mask*, ktorý špecifikuje cestu k maske defektov kvôli štatistickému vyhodnoteniu výsledných detekcií. Masky defektov je podrobnejšie rozoberaná v kapitole 4.1. Parameter *-ignore* udáva cestu k maske ignorovaných oblastí obrázku, ktoré nemajú žiadny vplyv na analýzu. Analyzovaná oblasť je v maske čierna. Príklad spustenia programu je ukázaný na obr. 3.1. Ukážka masky statických častí na obr. 3.2. Základnými funkciami celého algoritmu sú *drops\_detect\_sobel*, *drops\_detect\_difference* a *validate\_defect*, ktoré sú popísané v nasledujúcom texte.

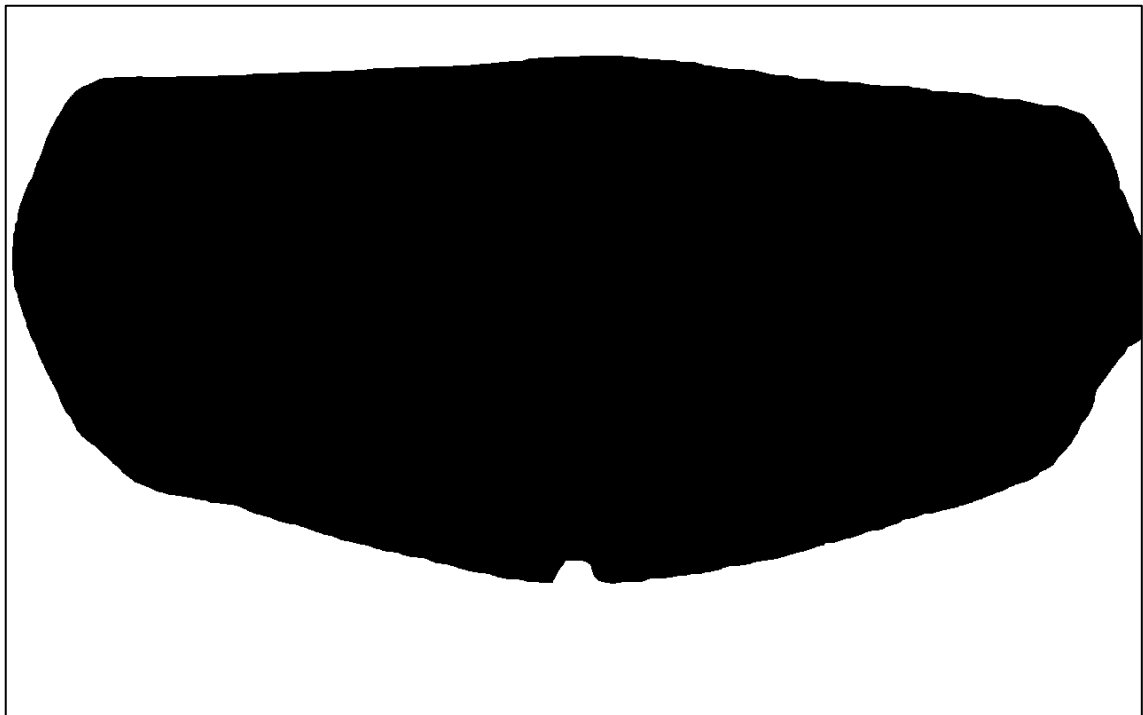
---

<sup>1</sup> <https://opencv.org/>

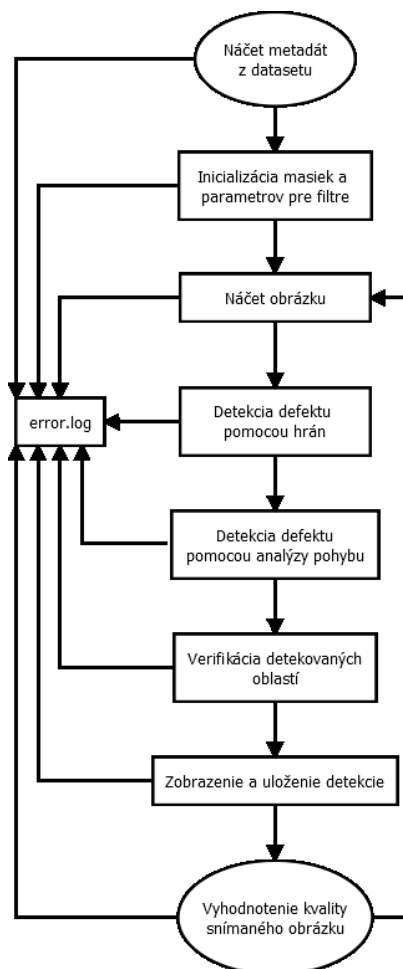
<sup>2</sup> <https://www.numpy.org/>

```
svato@svato-Aspire-VN7-792G ~  
File Edit View Search Terminal Help  
svato@svato-Aspire-VN7-792G ~ $ python bakalarka_release.py --input /path_to_dataset/  
--mask /path_to_mask/mask.png --ignore /path_to_ignore/ignore.png
```

*Obr. 3.1 Spustenie programu*



*Obr. 3.2 Maska ignorovaných oblastí*

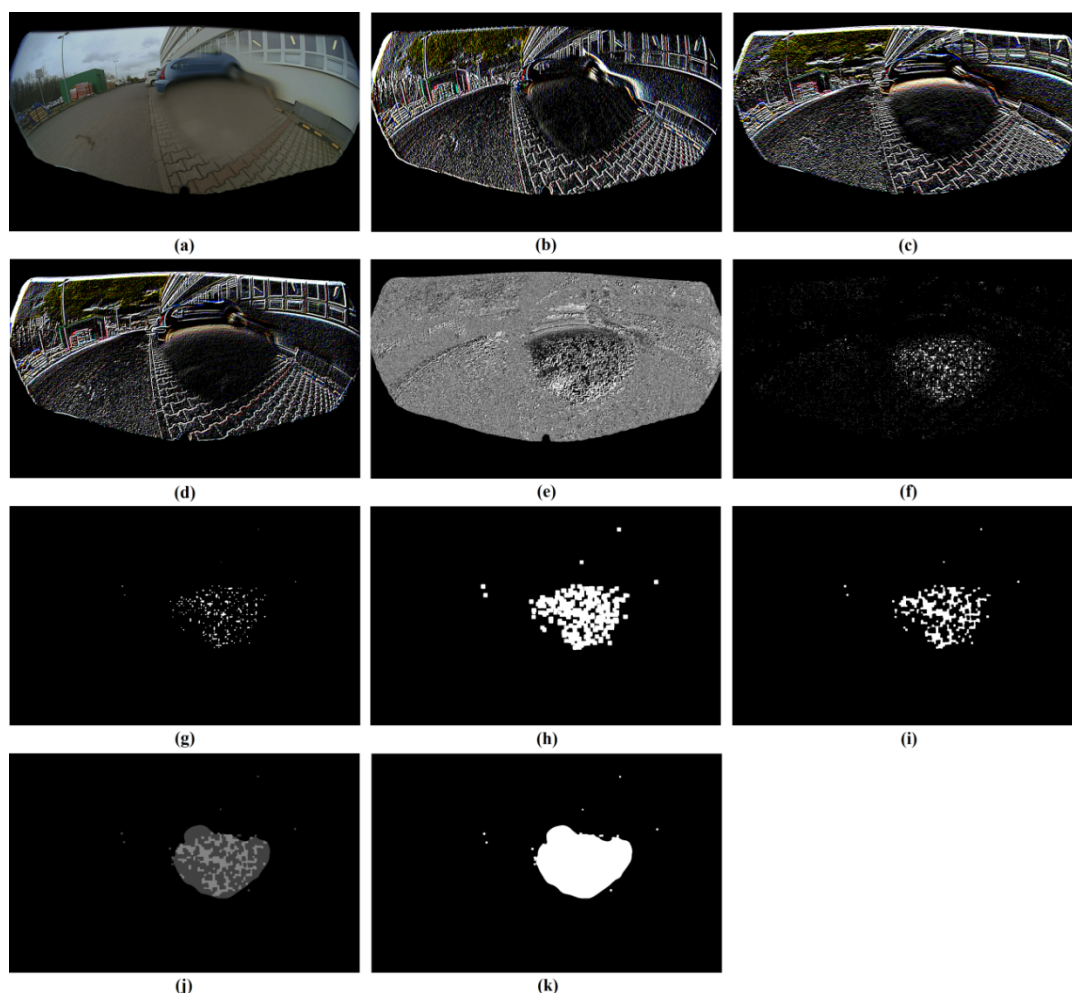


Obr. 3.3 Vývojový diagram programu

Funkcia *drops\_detect\_sobel* detekuje oblasti bez hrán Sobelovým filtrom vo vertikálnom (obr. 3.4(b)) i horizontálnom smere (obr. 3.4(c)). Základným predpokladom je buď rozmazanie vstupu po prechode vodnou kvapkou alebo defekt s jednoliatou farbou, čo znamená, že keď mám na kamere blato prípadne iné nečistoty, tak predpokladám, že oblasť nečistoty sa čo do hrany javí ako bezhranná. Následne sa detekcie hrán v oboch smeroch navzájom priemerujú. Výstup operácie priemerovania je zobrazený na obr. 3.4(d). Získaný obrázok sa prevedie do šedotónovej reprezentácie obr. 3.4(e), ktorá je potom binárne práhovaná obr. 3.4(f). Binárna reprezentácia hrán obsahuje veľké množstvo šumu. Šum je filtrovaný mediánovým filtrom s veľkosťou okna zadávanou užívateľom ako vstupný parameter funkcie, ktorého výstup je zobrazený na obr. 3.4(g). Jedná sa o zhluk malých oblastí bielych pixelov v oblasti defektu. Pre zvýraznenie oblasti defektu sú po sebe použité OpenCV funkcie *dilate* (obr. 3.4(h)) a *erode* (obr. 3.4(i)). Vzhľadom na kvazistatickosť defektu sa následne detekované oblasti bez hrán cez seba priemerujú (obr. 3.4(j)). Priemerovanie

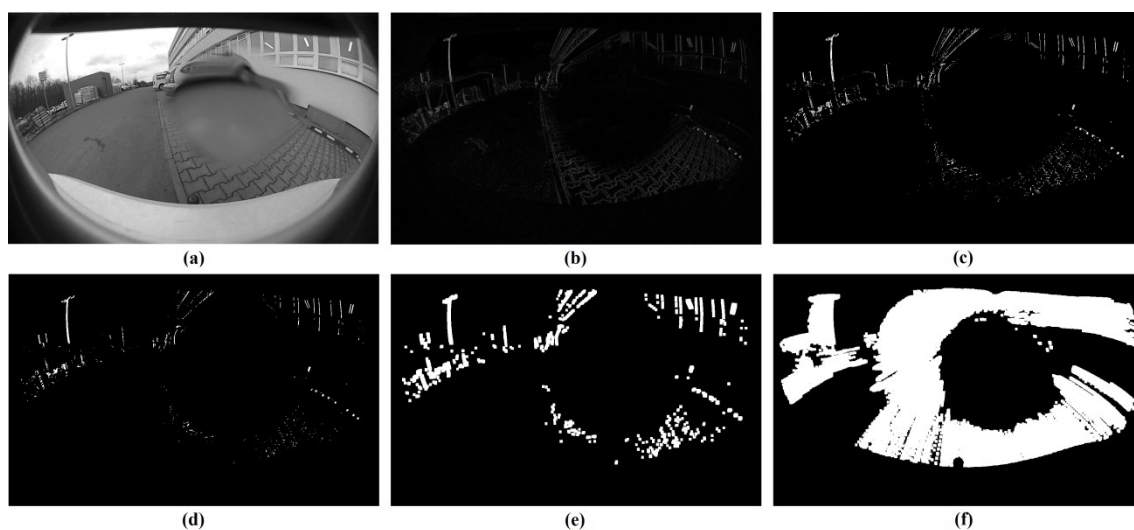
detekovaných oblastí eliminuje chybné detekcie. Posledným krokom je zvýraznenie defektu po priemerovaní aktuálnej a predchádzajúcej detekcie hrán (obr. 3.4(k)).

Vstupnými parametrami funkcie sú: predchádzajúca detekcia, aktuálna snímka, cesta k priečinku určenému na uloženie výstupu z funkcie, typ matematickej operácie – priemer alebo logický súčet, veľkosť okna určeného k filtrácii náhodného šumu a maska ignorovaných oblastí obr. 3.2. Funkcia porovnáva vývoj detekovaných oblastí v čase. Čím k väčšiemu posunu miesta defektu došlo tým menší dopad má daný defekt na celkovú kvalitu snímku. Samotná detekcia spočíva vo vyhľadávaní homogénnych oblastí. Čím plynulejší prechod a teda menší počet hrán v danom mieste snímky, tým väčšia pravdepodobnosť defektu. Výstupom funkcie je potom binárny obrázok samotných detekcií obr. 3.4(k).



*Obr. 3.4 Sled operácií s obrázkom pre funkciu drops\_detect\_sobel*

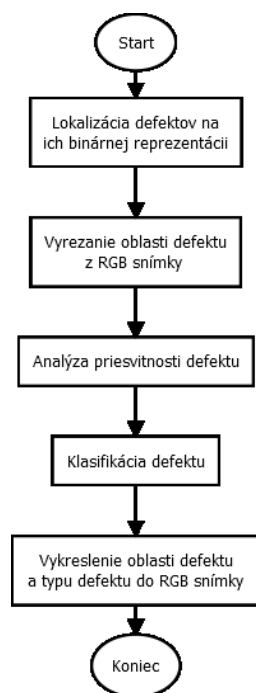
Funkcia *drops\_detect\_difference* implicitne hľadá nepohyblivé oblasti na scéne. Základným princípom funkcie je porovnávanie odlišnosti hodnoty jasu dvoch za sebou idúcich snímok. RGB reprezentácia je transformovaná do HSV z ktorého je vybratá posledná hodnota *value* reprezentujúca tmavosť/svetlosť pixelov v danej snímke obr. 3.5(a). Algoritmus je navrhovaný za predpokladu, že pri danom defekte môže dochádzať k zmene farby, napr. pokiaľ cez kvapku prechádzajú lúče z dvoch rozdielne farebných oblastí, ale hodnota jasu by mala byť zachovaná. Rozdiel aktuálnej a predchádzajúcej snímky po HSV transformácii je zobrazený na obr. 3.5(b), ktorá je potom binárne práhovaná (obr. 3.5(c)). Z binárnej reprezentácie defektu je následne odfiltrovaný šum, opäť mediánovým filtrom s veľkosťou okna zadávanou ako vstupný parameter funkcie (obr. 3.5(d)). Po odfiltrovaní šumu nasleduje zvýraznenie oblasti defektu použitím OpenCV funkcie *dilate* (obr. 3.5(e)). Výstupom funkcie je logický súčet sekvencie za sebou idúcich obrázkov (obr. 3.5(f)). Vstupnými parametrami sú opäť binárny obrázok predchádzajúcej detekcie, minulé snímka, aktuálny obrázok, miesto uloženia výstupu a veľkosť filtračného okna.



Obr. 3.5 Sled operácií s obrázkom pre funkciu *drops\_detect\_difference*

Ku klasifikácii defektov je použitá funkcia *validate\_defect*. Funkcia na základe binárnej reprezentácie, defekt lokalizuje a analyzuje na základe jeho RGB reprezentácie. Výsledná klasifikácia sa stanovuje pomocou výpočtu priemernej hodnoty RGB reprezentácie lokalizovaného defektu. Jedná sa o vyhodnotenie priesvitnosti defektu. Bahno prepúšťa menej svetla ako kvapka a teda oblasť na výslednom snímku je výrazne tmavšia ako v prípade vodnej kvapky. Pokiaľ sa jedná o hodnotu pod užívateľom nastaveným číslom, tak sa defekt vyhodnotí ako *dirt* – blato a vypíše sa do vstupného

obrázku inak sa vypisuje *drop* – kvapka. Funkcia zvýrazňuje detekované oblasti farebným štvorcom s popisom daného defektu. Farba ohraničenia štvorca udáva metódu použitú k detekcii defektu. Vývojový diagram funkcie *validate\_defect* je zobrazený na obr. 3.6.

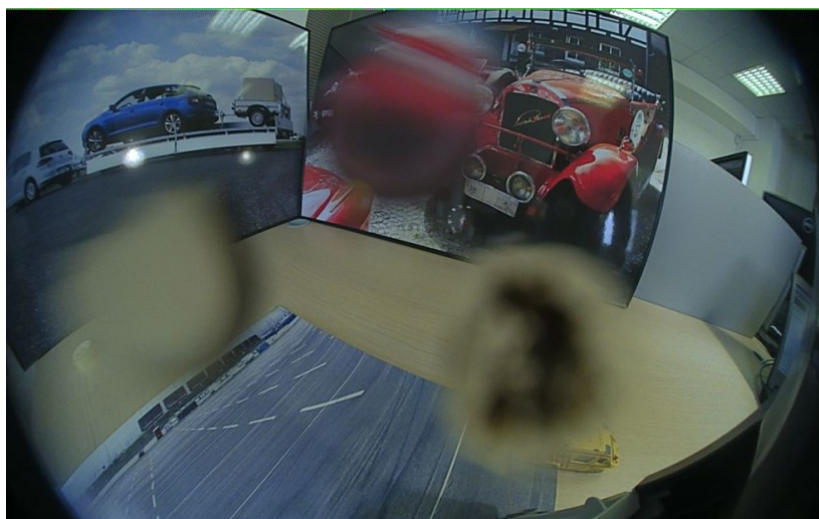


Obr. 3.6 Vývojový diagram funkcie *validate\_defect*

### 3.1.1. Vstup algoritmu

Algoritmus bol ladený na datasetoch obrázkov veľkosti 1280 x 800 pixelov. Dataset obsahuje postupnosť obrázkov zo samotného motorového vozidla, prípadne postupnosť obrázkov z prostredia kancelárie, kde boli modelované základné defekty a to rôzne veľkosti vodných kvapiek, bahna a ich kombinácii na optickom systéme kamery. Jedná sa o smart kameru s 1 megapixelovým rozlíšením, 30 FPS, ktorá býva štandardne použitá vo vozoch koncernu Volkswagen. Pri simulácii defektu boli postupne pridávané vymenované typy znečistenia na šošovku kamery. S kamerou bolo nasledovne hýbané v priestore a následne boli analyzované zozbierané dáta. Kamera je zhodná s kamerou umiestnenou na motorovom vozidle. Po čiastočnej analýze testujúcich dát z kancelárie boli vyšpecifikované konkrétne požiadavky na testovanie použitím motorového vozidla. Dáta zozbierané z jazdy na aute obsahujú záznam bez defektu, záznam s vodnými kvapkami a bahnom. Na kameru motorového vozidla boli postupne pridávané rôzne typy nečistôt o rôznej veľkosti. Defekty sú navrhované tak aby dáta obsahovali rôzne

kombinácie defektov a ich veľkostí pre lepší odhad použitia a funkcionality navrhnutého algoritmu. Dáta z oboch meraní boli použité na overenie správnosti detekcie navrhnutého algoritmu. Ukážky vstupných snímok z oboch prostredí sa nachádzajú na obr. 3.7 a obr. 3.8. Algoritmus bol ladený na 8 rôznych typoch datasetov s celkovým počtom 9468 snímok. Testovanie programu prebiehalo na 28 záznamoch z motorového vozidla. Hodnoty prahov a filtrov boli zvolené experimentálne a to konkrétne prah pri funkcii *drops\_detect\_difference* je rovný 50 (obr. 3.5(c)), pri funkcii *drops\_detect\_sobel* je rovný 5 obr. 3.4(f). Veľkosť okna pri mediánových filtroch je zvolená na 3 pri *drops\_detect\_difference* (obr. 3.5(d)) a 5 pri *drops\_detect\_sobel* (obr. 3.4(g)). Pri funkcii *validate\_defect* je číslo klasifikujúce defekt nastavené na hodnotu 50.



*Obr. 3.7 Príklad simulovaného defektu*



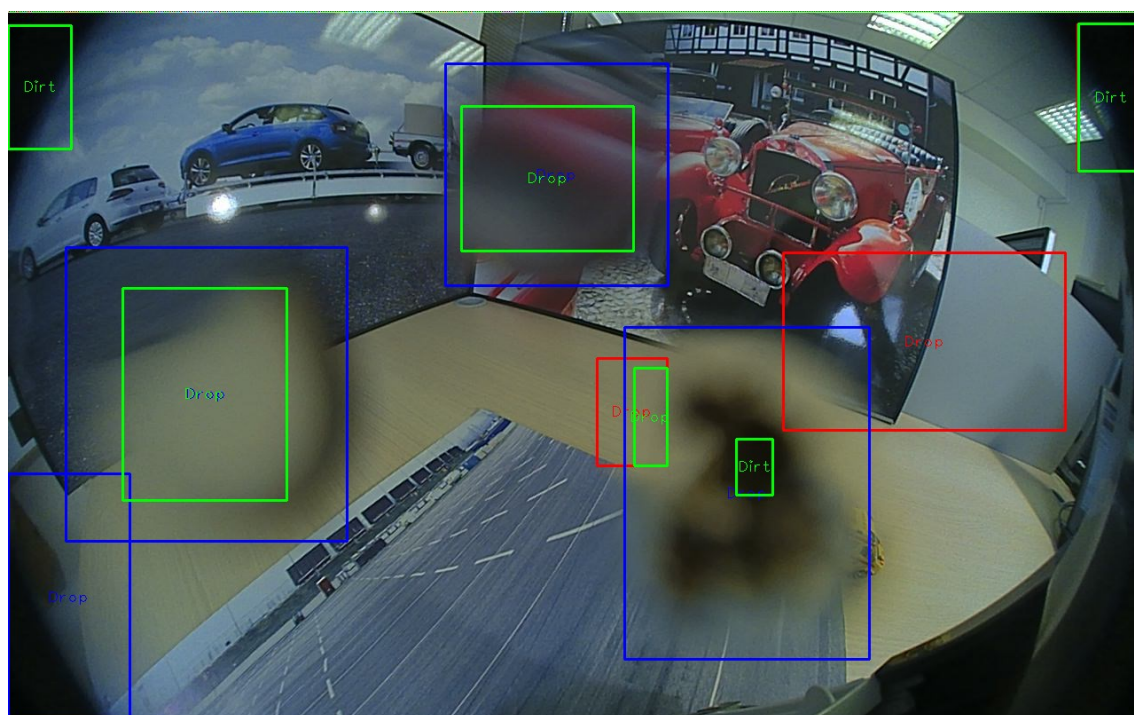


Obr. 3.8 Príklady reálnych defektov

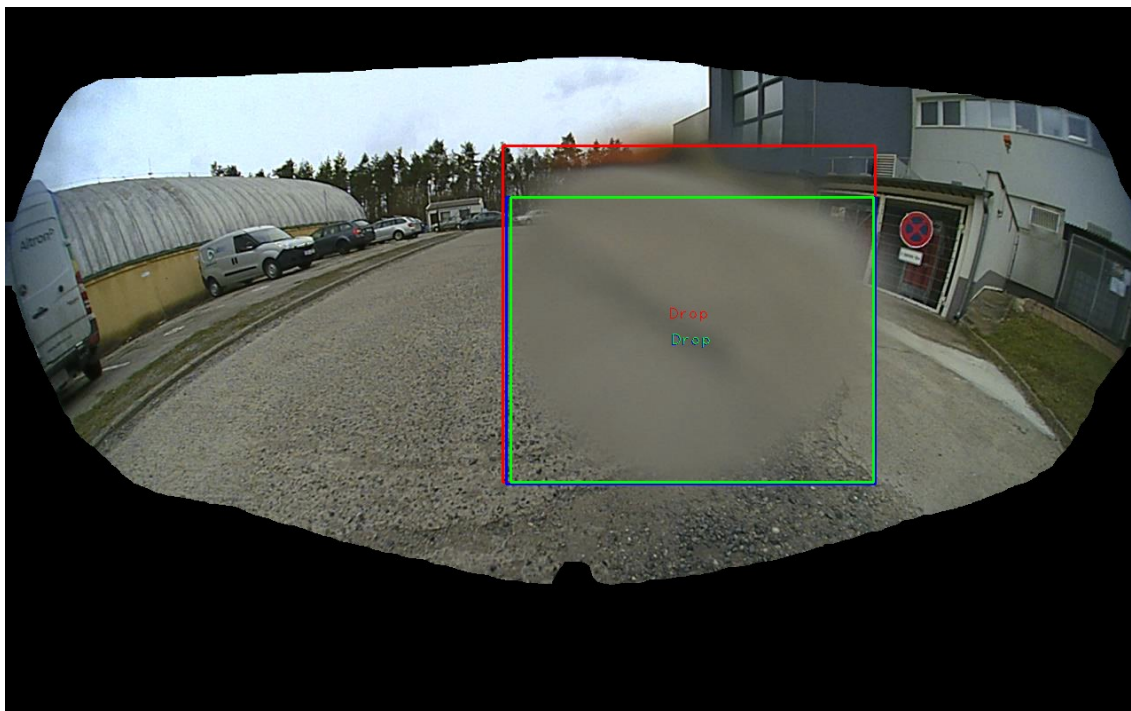
### 3.1.2. Výstup algoritmu

Úlohou bakalárskej práce bolo navrhnúť a otestovať spoľahlivosť algoritmu, ktorý vyhodnocuje defekty kamery za účelom stanovenia spoľahlivosti daného optického aparátu. Algoritmus klasifikuje defekty do dvoch kategórií a to priehľadná / nepriehľadná vada. Výstupná detekcia je zobrazená na obr. 3.9, obr. 3.10 a obr. 3.11. Klasifikácia je rozdelená do troch typov. **Modré** zvýraznenie defektu značí, že defekt je detekovaný pomocou funkcie *drops\_detect\_difference*, **červená** farba reprezentuje použitie *drops\_detect\_sobel* filtru. Najdôležitejšia farba je **zelená**, ktorá indikuje verifikovaný defekt, tzn. defekt, ktorý je detekovaný použitím oboch metód detekcie. Medzi výstupné veličiny patrí výstup z funkcie *drops\_detect\_difference* zobrazený obr. 3.5(f). Výstup z funkcie *drops\_detect\_sobel* je zobrazený na obr. 3.4(k). Pri funkcii *drops\_detect\_difference* je nutné si všimnúť inverziu farieb. Funkcia postupne vybieli výstupnú masku oblasťami, pri ktorých dochádza k pohybu. Pričom *drops\_detect\_sobel* priamo znázorňuje oblasti bez hrán ako biele objekty v maske detekcií. Medzi výstup patrí i percento vyjadrujúce spoľahlivosť / zašpinenie šošovky kamery. Užívateľ si môže vybrať či chce počítať spoľahlivosť zo všetkých detekcií alebo len z potvrdených.

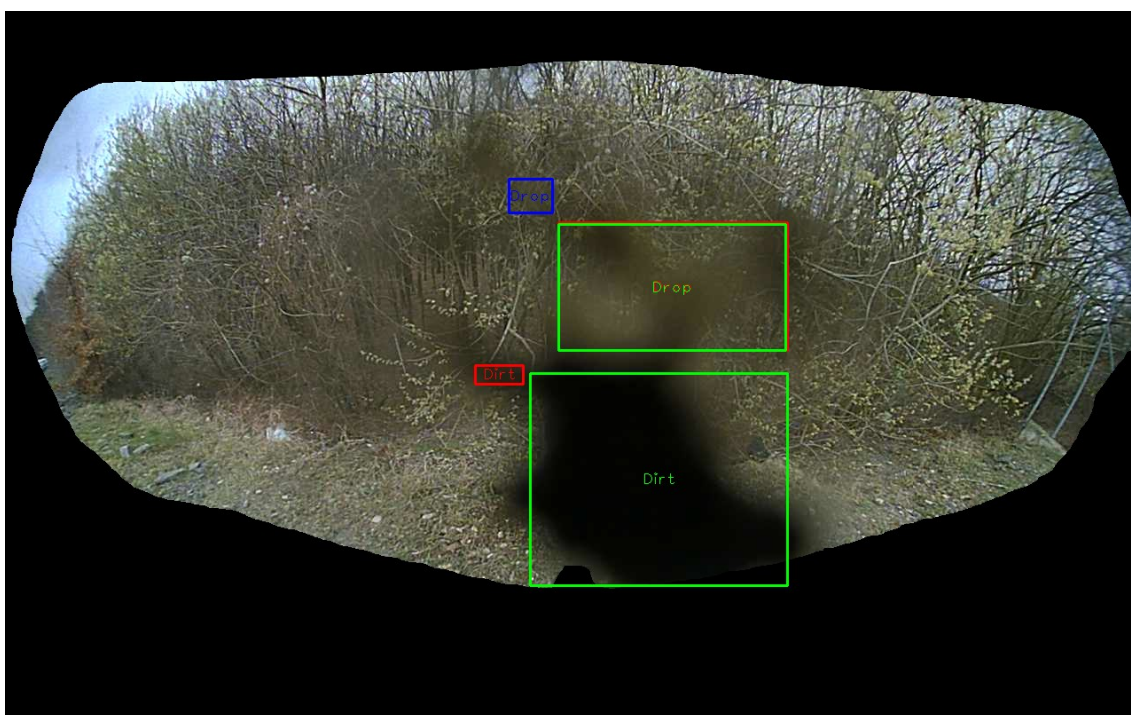
V našom prípade sa stanovuje spoľahlivosť pre každú snímku zo všetkých detekcií a ukladá do výstupného textového súboru (*A\_fit.txt*). Algoritmus ukladá výsledky do zadaného priečinku vstupu. Jednotlivé detekcie a vyhodnotenia sú rozlíšiteľné podľa názvu výstupného obrázka. V prípade chyby sa do logovacieho súboru ukladajú chybové kódy programu. Algoritmus generuje textové súbory určené k vyhodnoteniu správnosti detekcie programu a jeho rýchlosti. Vyhodnotenie kvality programu je popísané v nasledujúcej kapitole. V reálnom nasadení bude algoritmus predávať informáciu o tom na koľko je znečistený optický systém vozidla a v prípade potreby upozorní o nutnosti vyčistenia prípadne o tom že spoľahlivosť autonómneho systému riadenia je znížená. Vzhľadom na nutnosť jednoduchosti a rýchlosti daného algoritmu k ukladaniu výsledkov detekcií už dochádzať nebude.



*Obr. 3.9 Klasifikácia defektu pri jeho simulácii*



*Obr. 3.10 Klasifikácia defektu pri reálnom použití - kvapka*



*Obr. 3.11 Klasifikácia defektu pri reálnom použití - bahno*



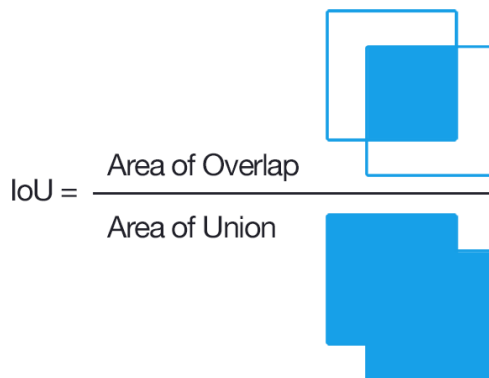
## 4. VYHODNOTENIE NAVRHNUTÉHO ALGORITMU

Vyhodnotenie správania algoritmu zahrňuje analýzu Intersection over Union (IoU), ktorá bude popísaná v nasledujúcom texte, analýzu vzniknutého šumu pri danej metóde, stanovenie priemernej hodnoty IoU na daný záznam, stanovenie času učenia ( $T_0$ ) danej metódy čo odpovedá dosiahnutiu 50% IoU a vyhodnotenie priemernej rýchlosti pre jednotlivé metódy analýzy. K vyhodnoteniu bolo použitých 28 záznamov z kamery vozidla. Vzhľadom na objem dát sú v práci popísané len vybrané výsledky analýz, ostatné výsledky sú dodané na priloženom médiu. Tabuľka 4.1 zahrňuje hodnoty analyzovaných veličín vybraných záznamov.

### 4.1. Intersection over Union (IoU)

IoU je vyhodnocovacia metrika používaná k stanoveniu presnosti detekcie objektu pre daný dataset. Na to aby mohla byť metóda IoU požitá na vyhodnotenie detekcie objektu, je nutné stanoviť *ground-truth*, čo reprezentuje ručne špecifikovanú oblasť defektu a detekciu objektu analyzovaným programom. Ručne stanovená poloha defektu je zobrazená na obr. 4.2(b) pričom jeho detekcia funkciou *drops\_detect\_sobel* je zobrazená na obr. 4.2(c). [12]

Základný princíp metódy je vysvetlený na nasledujúcom obrázku, obr. 4.1. V čitateli sa nachádza oblasť prekryvu, ktorá je v našom prípade počítaná pomocou logického *and* *ground-truth* s detekovanou oblasťou defektu. Menovateľ obsahuje celkovú plochu oboch masiek reprezentovaných logickým *or*. Výsledná hodnota IoU stanovuje ako presne je detekovaný defekt v prekrytí s ručne vyznačeným defektom. [12]



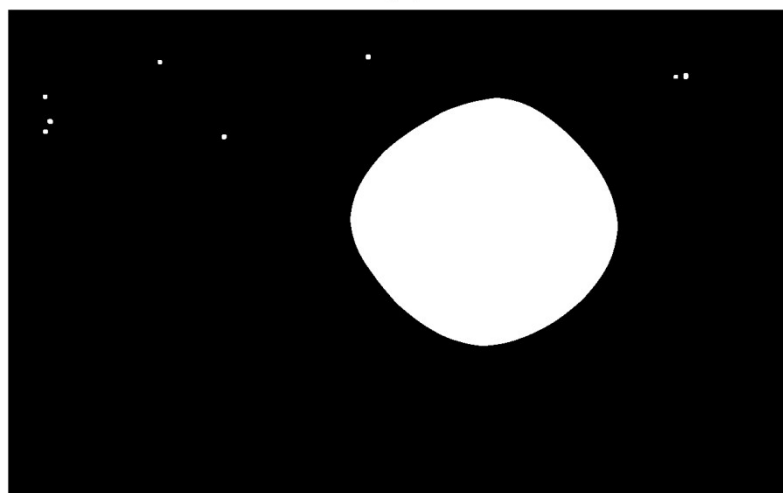
Obr. 4.1 Princíp výpočtu IoU [12]



(a)



(b)

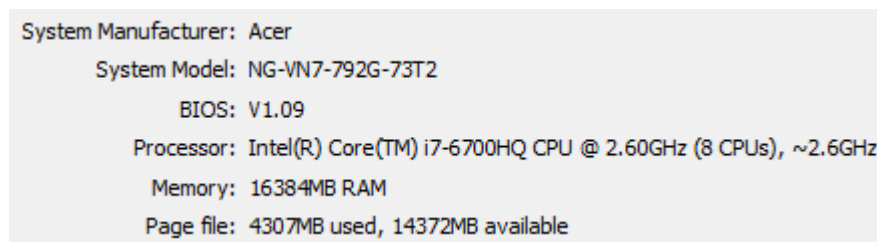


(c)

Obr. 4.2 Vstupy IoU

## 4.2. Vyhodnotenie priemernej rýchlosti algoritmu

Stanovenie rýchlosti analýzy je kľúčové pri použití v embedded svete. Je dôležité aby metódy analýzy boli čo možno najrýchlejšie. Rýchlosť výpočtu závisí na hardware-ovom vybavení daného prístroja. Špecifikácie počítača, na ktorom bol program spustený sú zobrazené na obr. 4.3. Analýza na grafickej karte prípadne v multijadrovom prevedení nebola použitá. Priemerná rýchlosť analýzy jednej snímky funkcie *drops\_detect\_sobel* je 0.265 s pričom funkcia *drops\_detect\_difference* má priemernú rýchlosť rovnú 0.035 s. Rozdiel v rýchlosti výpočtu je daný matematickou náročnosťou danej metódy. V momentálnej fáze projektu nebol kladený dôraz na rýchlosť výpočtu. Optimalizácia rýchlosti je zatiaľ viazaná len na výber časovo nenáročných metód detekcie defektu. Rýchlosť je ďalej možné optimalizovať napr.: znížením rozlíšenia vstupnej snímky, prípadne jej transformácia do šedotónovej reprezentácie. Rýchlosť je možné zvýšiť i znížením frekvencie vyhodnotenia snímky. Momentálne sa vyhodnocuje každá snímka.



Obr. 4.3 Špecifikácie vyhodnocovacej jednotky

## 4.3. Vyhodnotenie detekcie algoritmu

V nasledujúcom texte bude popísané vyhodnotenie algoritmu na referenčnom zázname bez nečistôt, vyhodnotenie detekcie priehľadných nečistôt a vyhodnotenie detekcie nepriehľadných nečistôt. K vyhodnoteniu je použitá metrika IoU na analýzu prekrytia detekcie s *ground-truth* a analýza šumu.

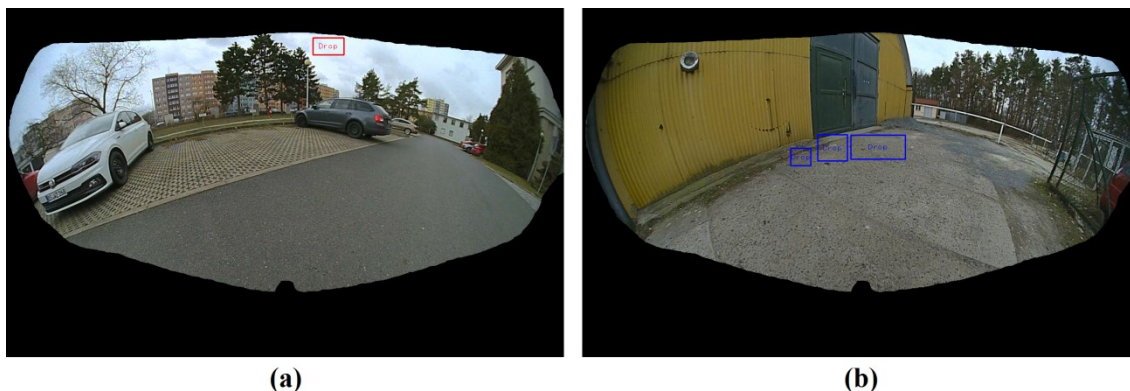
Šum je vypočítaný ako pomer falošnej detekcie nečistoty v oblasti mimo *ground-truth* (obr. 4.2(b)) k veľkosti snímky.

### 4.3.1. Vyhodnotenie navrhnutých metód pri referenčnom zázname

Referenčný záznam je záznam, ktorý neobsahuje žiadne nečistoty na optickom aparáte kamery. Detekcie, ktoré vznikajú sú falošné detekcie dané chybou metódy. Funkcia *drops\_detect\_sobel* generuje falošne pozitívne detekcie v prípade detekcie akýchkoľvek

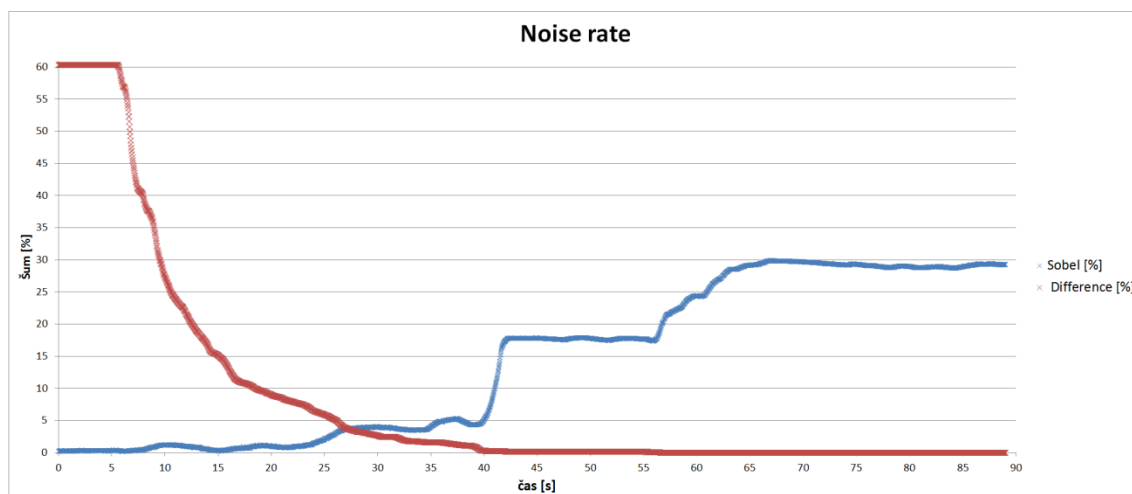
homogénnych oblastí. Bezhranné oblasti môžu byť steny budov, dvere áut, prípadne veľkým zdrojom chybných detekcií v prípade funkcie *drops\_detect\_sobel* je obloha, ktorej ukážka je zobrazená na obr. 4.4(a). Obloha obsahuje malý počet hrán a i matematickým popisom sa približuje vlastnostiam vodnej kvapky.

Vzhľadom na princíp vyhodnotenia defektu funkcie *drops\_detect\_difference* vysvetlený na obr. 3.5, vzniká veľký počet falošne pozitívnych detekcií na začiatku analýzy, kým nie je metóda dostatočne natrénovaná. Problematické sú oblasti, ktoré na veľkej ploche snímky nesú určitú podobnosť. Príkladom takýchto oblastí sú trávnaté plochy, prípadne cesty. V takomto prípade je nutné počkať na stav, kedy sa charakter snímanej scény pozmení. Príklad falošne pozitívnej detekcie funkcie *drops\_detect\_difference* je zobrazený na obr. 4.4(b).

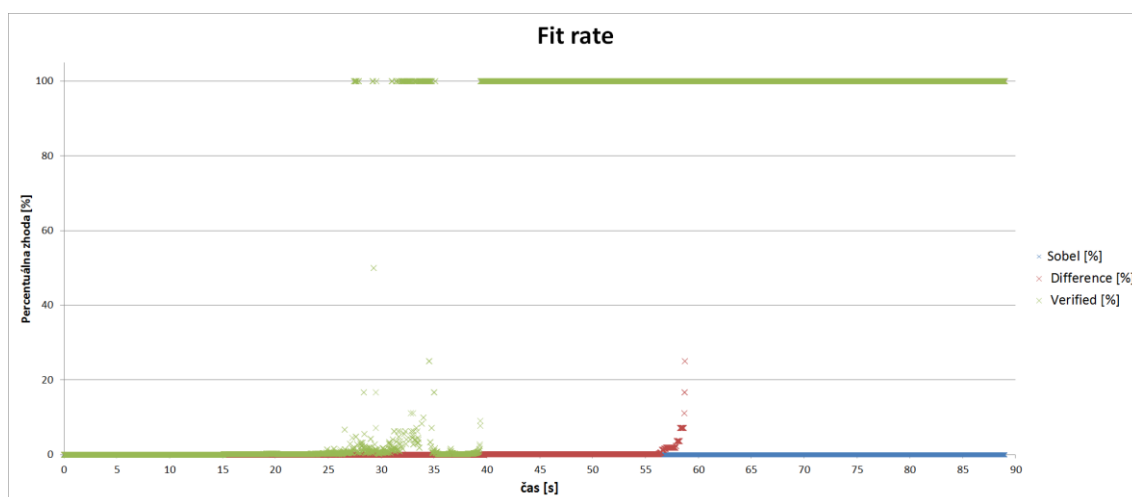


Obr. 4.4 Chybné detekcie metódy *drops\_detect\_sobel* (a) a *drops\_detect\_difference* (b)

Výsledky vyhodnotenia IoU a vyhodnotenia šumu zobrazuje graf 4.1 a graf 4.2. Čas učenia funkcie *drops\_detect\_difference* je rovný 58.76 s. pričom pri verifikovaných detekciách je  $T_0 = 27.43$  s. Pri funkcii *drops\_detect\_sobel* čas učenia nebol stanovený vzhľadom na veľký obsah šumu generovaného touto metódou. Graf 4.1 poukazuje na ustálenie hodnoty šumu funkcie *drops\_detect\_sobel* v okolí 30 %, čo je príliš vysoká hodnota na stanovenie  $T_0$ .



Graf 4.1 Vyhodnotenie hodnôt šumu referenčného záznamu v závislosti na dobe analýzy



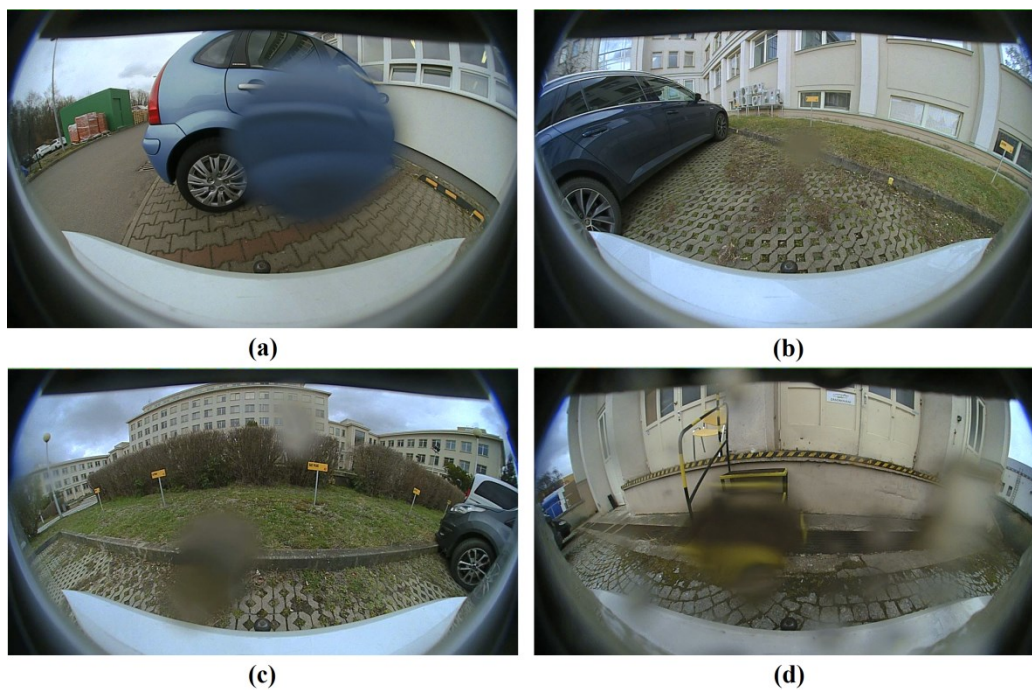
Graf 4.2 Vyhodnotenie hodnôt IoU referenčného záznamu v závislosti na dobe analýzy

### 4.3.2. Detekcia priesvitných nečistôt

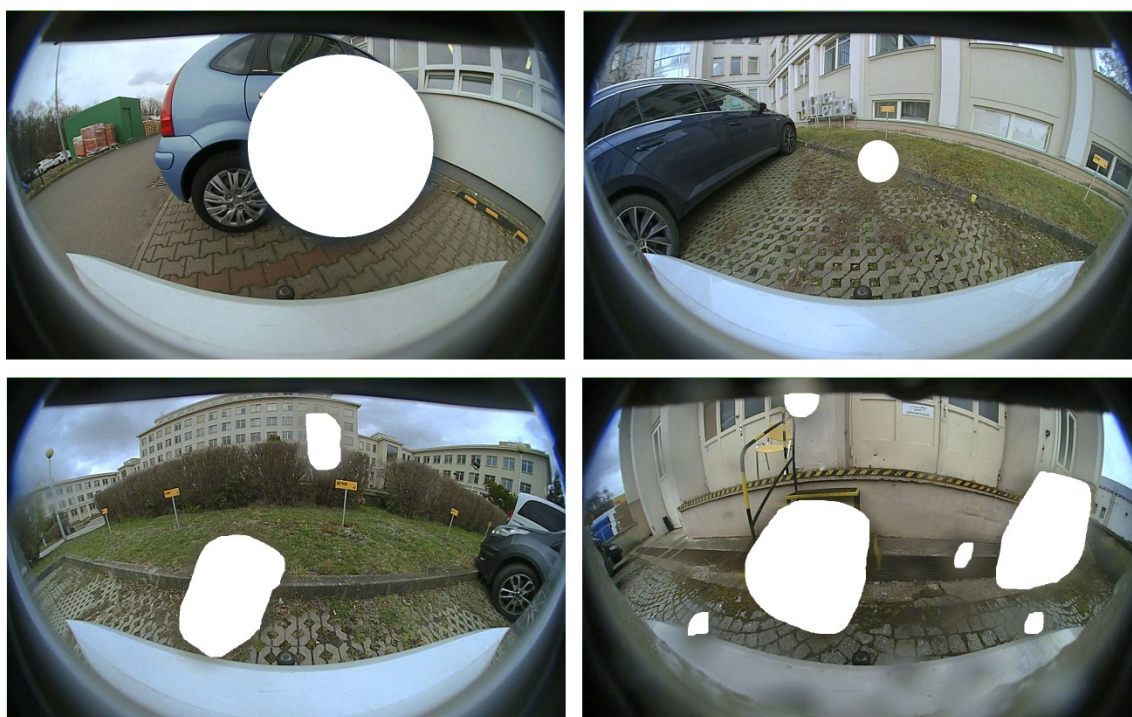
Hlavnou charakteristikou priesvitných nečistôt je rozmazanie snímky v oblasti nečistoty. Príklady defektov sú zobrazené na obr. 4.5 a ich *ground-truth* na obr. 4.6. Pri defekte zobrazenom na obr. 4.5(d) je nutné si všimnúť zložitosť stanovenia *ground-truth* masky vzhľadom na veľký počet defektov na snímke. Je zložité určiť, ktorá kvapka spôsobuje na toľko veľké rozmazanie, že je ju možné považovať za defekt.

Graf 4.3, graf 4.4, graf 4.5 a graf 4.6 zobrazujú výsledné vyhodnotenie IoU defektov priesvitných nečistôt.

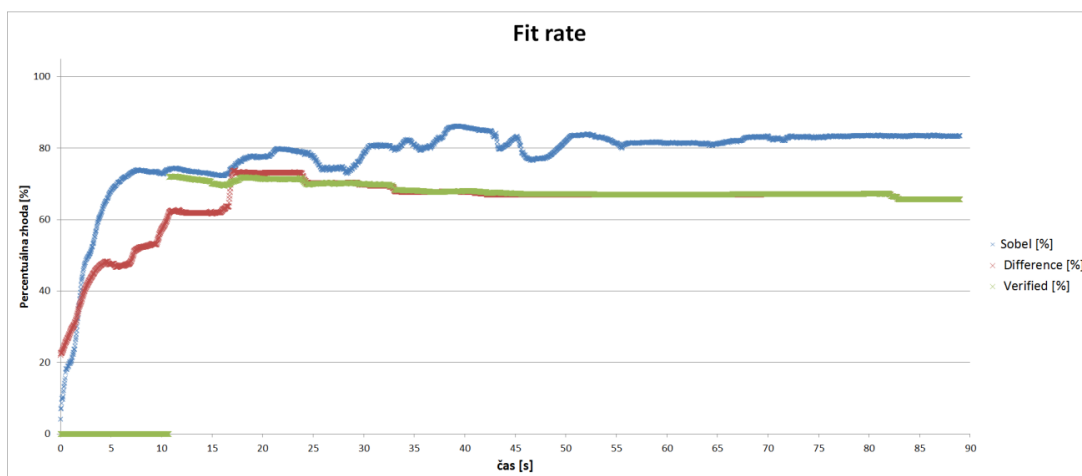




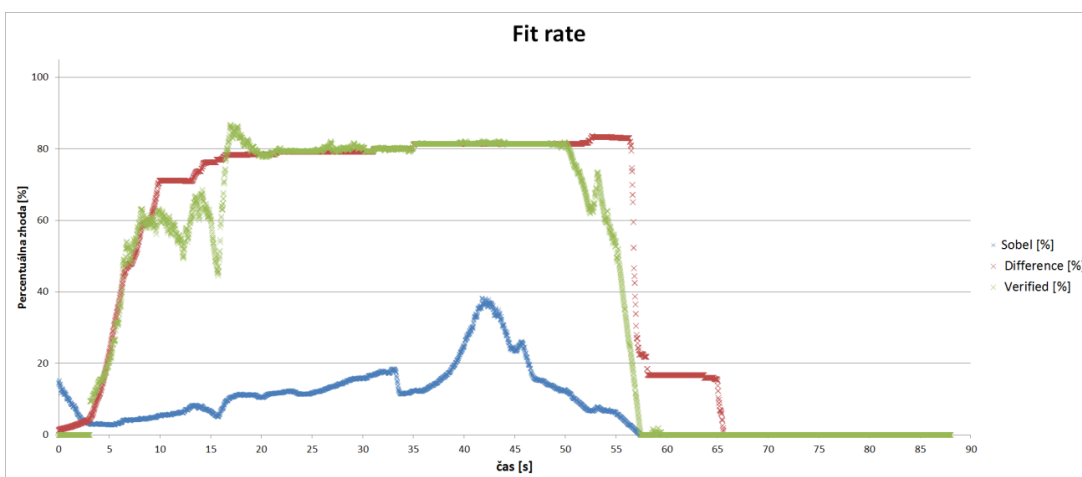
Obr. 4.5 Príklady defektov priesvitných nečistôt



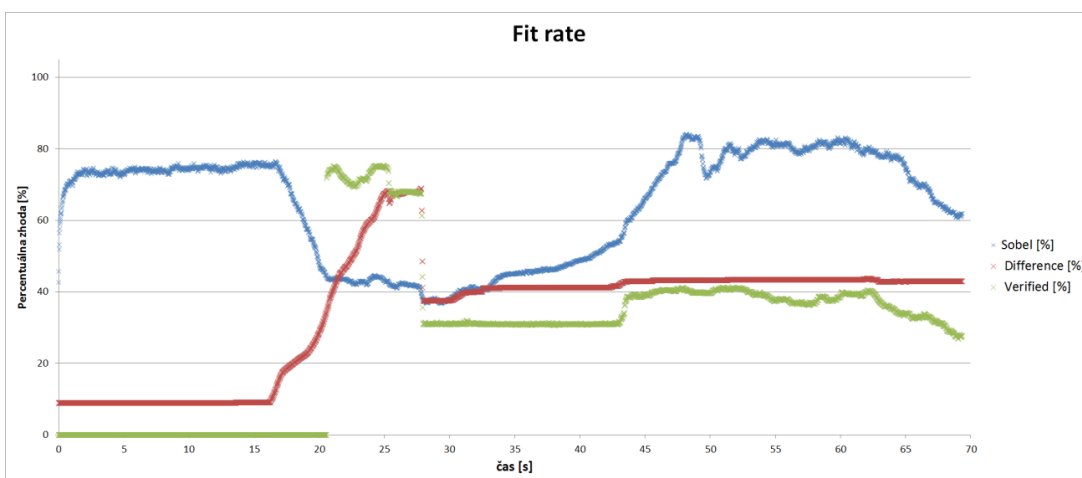
Obr. 4.6 Ground-truth masky priesvitných defektov



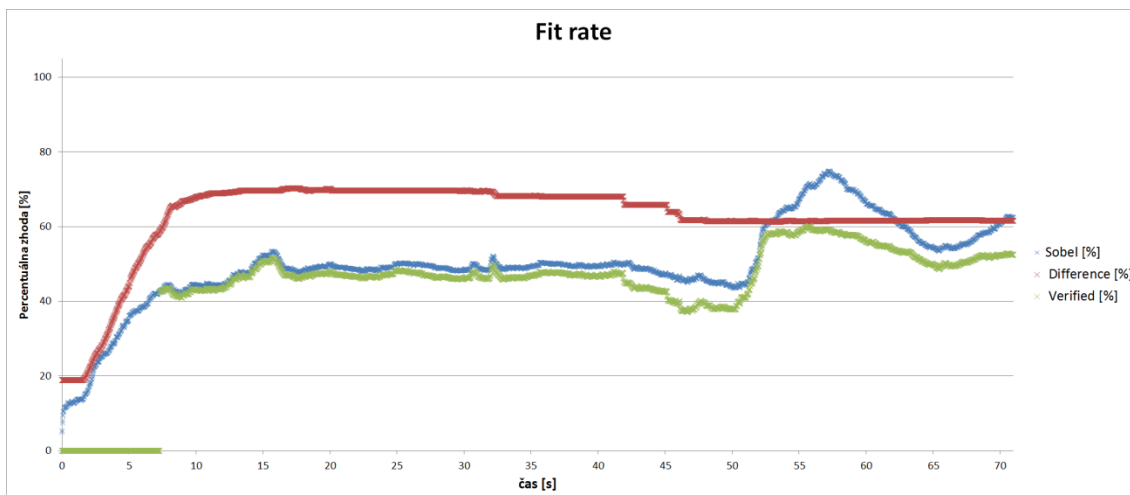
Graf 4.3 Vyhodnotenie IoU záznamu nečistoty z obr. 4.5(a) v závislosti na dobe analýzy



Graf 4.4 Vyhodnotenie IoU záznamu nečistoty z obr. 4.5(b) v závislosti na dobe analýzy



Graf 4.5 Vyhodnotenie IoU záznamu nečistoty z obr. 4.5(c) v závislosti na dobe analýzy



Graf 4.6 Vyhodnotenie IoU záznamu nečistoty z obr. 4.5(d) v závislosti na dobe analýzy

Graf 4.4 poukazuje na zníženie IoU detekcie funkciou *drops\_detect\_sobel* spôsobené šumom pochádzajúcim z oblasti oblohy prípadne stien budov v okolí dráhy vozidla. V cca 57 s dochádza k skokovému poklesu IoU, ktorý spôsobuje odparenie vodnej kvapky (obr. 4.7). Pričom graf 4.5 zobrazuje pokles spôsobený skokovou zmenou jasu snímky v oblasti defektu spôsobenou prechodom pre chodcov (obr. 4.8).



Obr. 4.7 Odparenie defektu obr. 4.5(b)



Obr. 4.8 Skoková zmena jasu v oblasti defektu obr. 4.5(c)

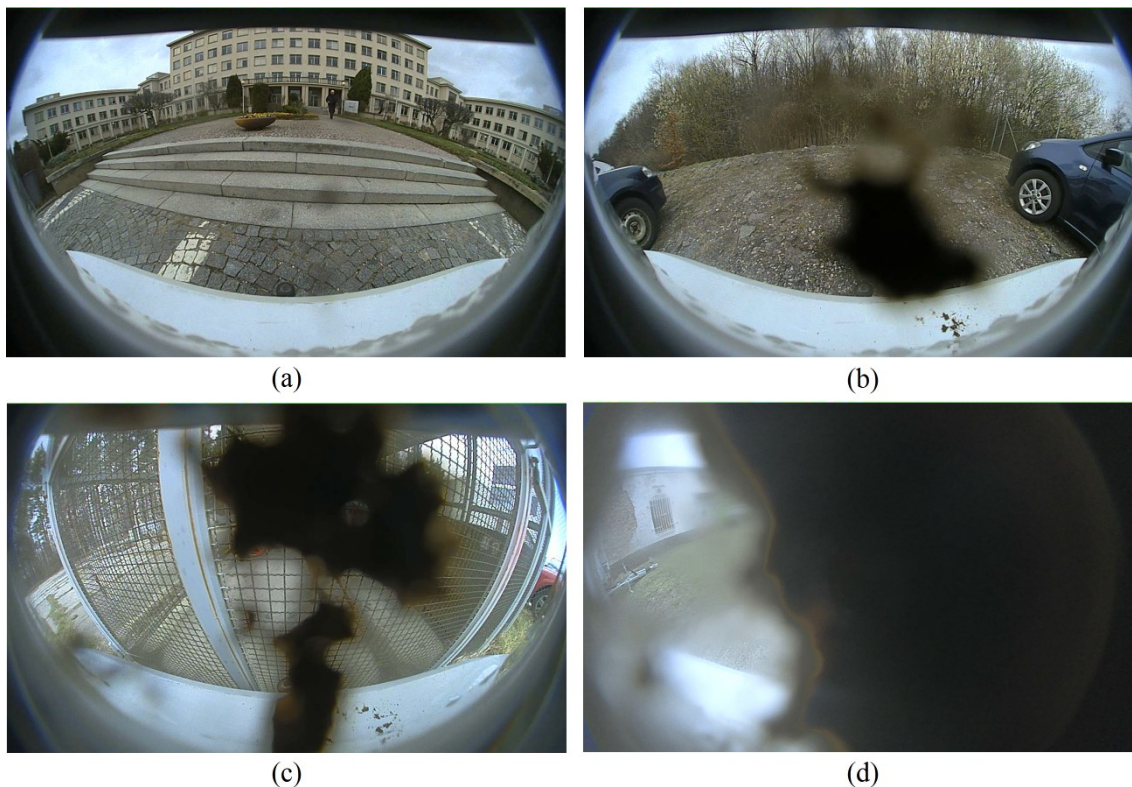


Na základe analýzy IoU prievitných defektov z obr. 4.5 je možné predpokladať, že funkcia *drops\_detect\_sobel* má rýchlejší nárast IoU pri nepohyblivých scenároch – kým sa vozidlo rozbehne. Metóda je náchylnejšia na chyby spôsobené jasnými oblasťami ako je obloha prípadne chyby spôsobené plochami bez hrán. *Drops\_detect\_difference* má pomalší nábeh každopádne je robustnejšia. Problém nastáva pri skokovej zmene farby prípadne jas v oblasti defektu. Táto skoková zmena je vyhodnotená ako pohyb čím zmenší plochu detekovaného defektu.

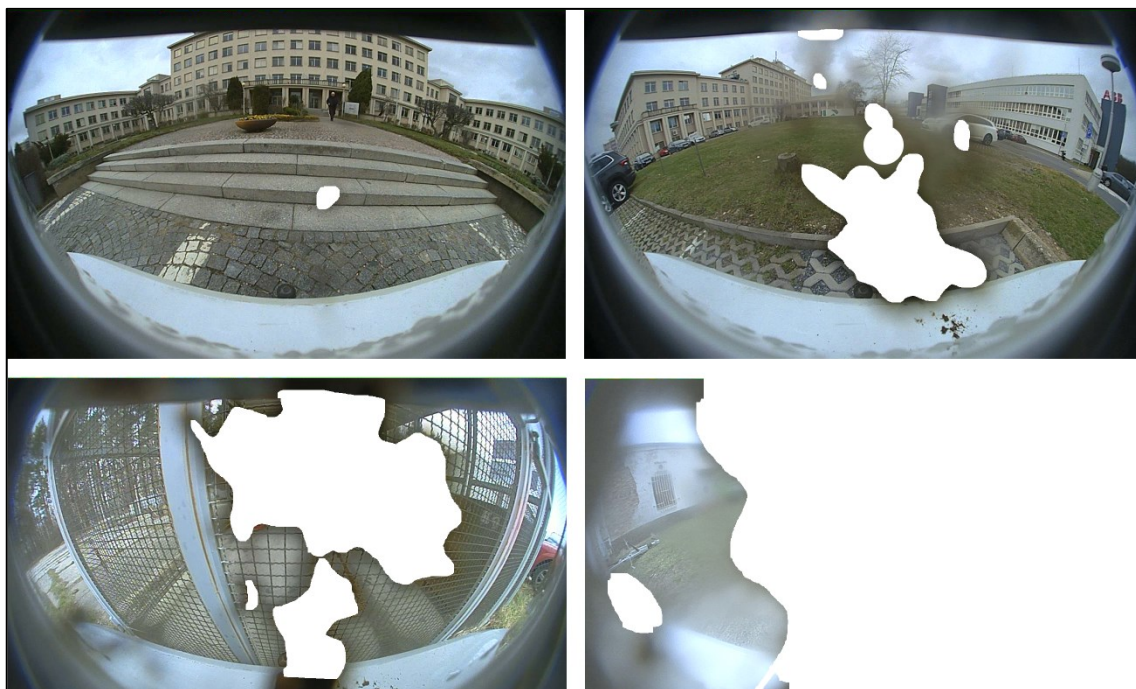
### 4.3.3. Detekcia neprievitných nečistôt

Hlavnou charakteristikou neprievitných nečistôt je zatmavenie snímky v oblasti nečistoty. Príklady defektov sú zobrazené na obr. 4.9 ich *ground-truth* na obr. 4.10. Pri defekte zobrazenom na obr. 4.9(d) je nutné si všimnúť zložitosť stanovenia *ground-truth* masky. Opäť je ťažké definovať ako veľmi môže byť nepriehľadná daná plocha snímky.

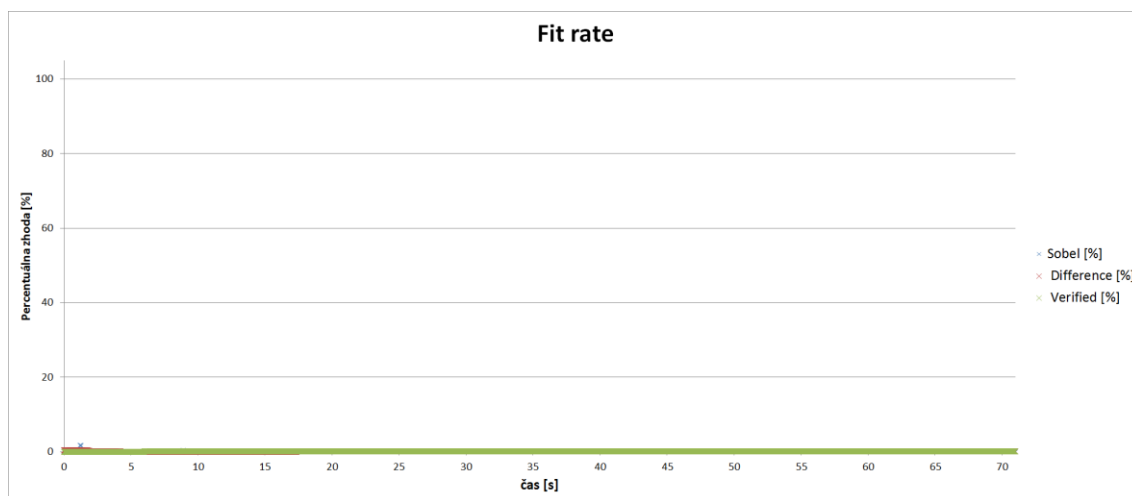
Graf 4.7, graf 4.8, graf 4.9 a graf 4.10 zobrazujú výsledné vyhodnotenie IoU neprievitných defektov.



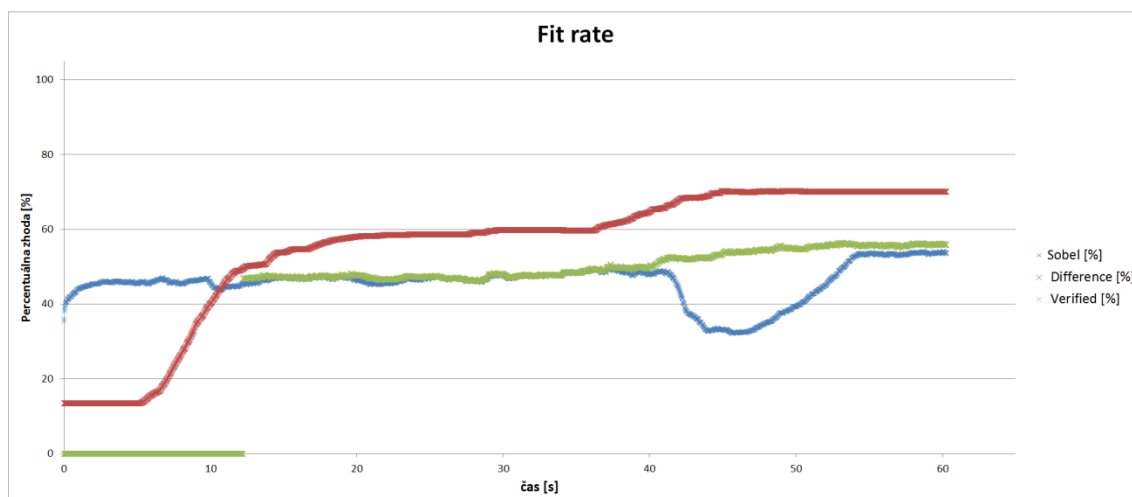
Obr. 4.9 Príklady defektov neprievitných nečistôt



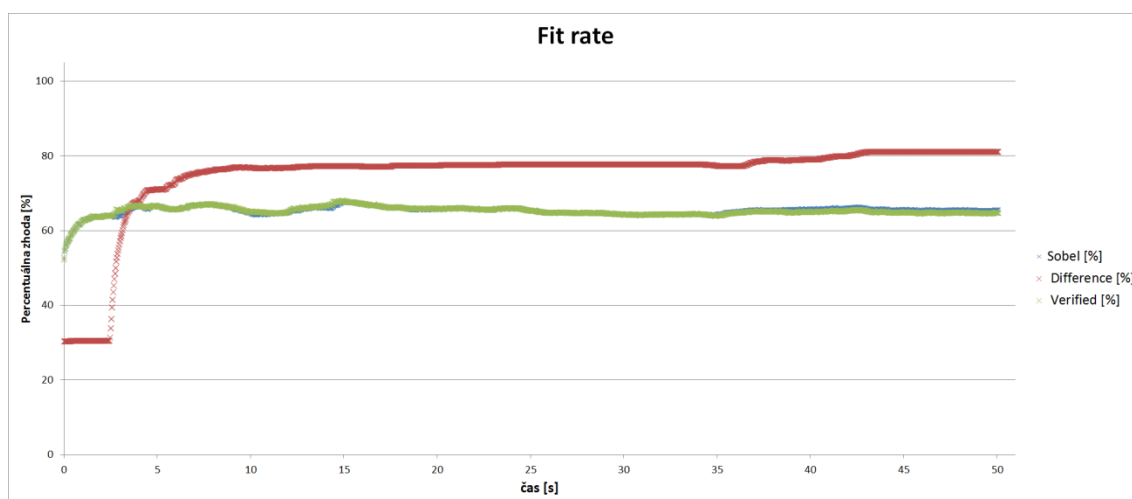
Obr. 4.10 Ground-truth masky nepriesvitných defektov



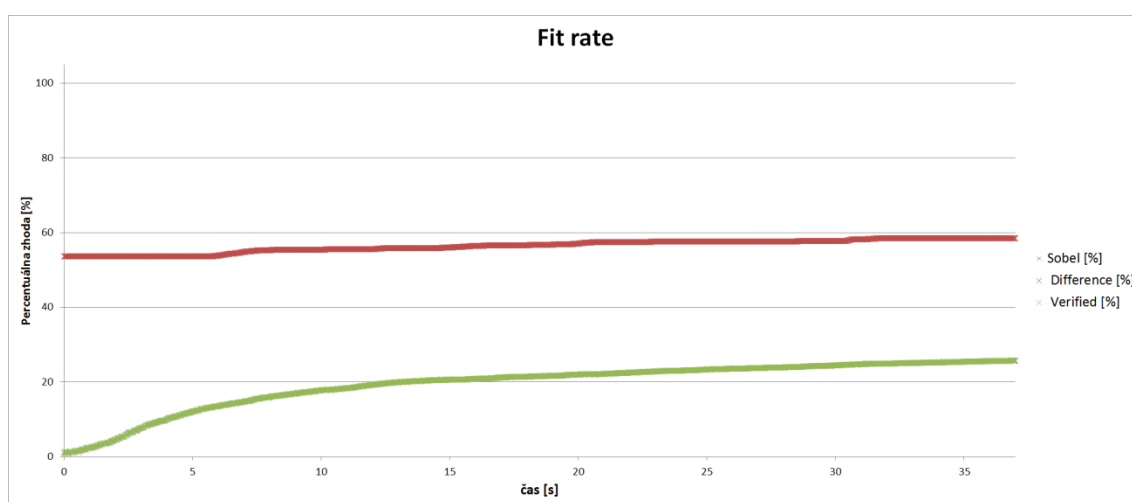
Graf 4.7 Vyhodnotenie IoU záznamu nečistoty z obr. 4.9(a) v závislosti na dobe analýzy



Graf 4.8 Vyhodnotenie IoU záznamu nečistoty z obr. 4.9(b) v závislosti na dobe analýzy



Graf 4.9 Vyhodnotenie IoU záznamu nečistoty z obr. 4.9(c) v závislosti na dobe analýzy



Graf 4.10 Vyhodnotenie IoU záznamu nečistoty z obr. 4.9(d) v závislosti na dobe analýzy

Z vyhodnotenia obr. 4.9(a) vyplýva, že nečistota podobného charakteru je príliš malá k detekcii. Obe metódy nedokážu defekt rozoznať. Je otázne, či daný typ defektu je možné odignorovať alebo je potrebné navrhnutie citlivejších metód detekcie.

Graf 4.8 v okolí 45 s poukazuje na pokles IoU funkcie *drops\_detect\_sobel*. Pokles je spôsobený falošnou detekciou karosérie dvoch áut zobrazenej na obr. 4.11.



Obr. 4.11 Falošná detekcia metódou *drops\_detect\_sobel*

Na základe analýzy IoU nepriesvitných defektov z obr. 4.9 je možné predpokladať, že funkcia *drops\_detect\_sobel* má opäť rýchlejší nárast IoU pri nepohyblivých scenároch – kým sa vozidlo rozbehne. Funkcia *drops\_detect\_difference* má pomalší nábeh každopádne je omnoho účinnejšia pri detekovaní nepriesvitných defektov.

Nasledujúca tabuľka zhrňuje základné parametre analýzy priesvitných i nepriesvitných defektov. **Modrá** farba odpovedá funkcii *drops\_detect\_sobel* a **červená** funkcii *drops\_detect\_difference*.

Defekt	T <sub>0</sub> [s]	Priemer IoU [%]	Defekt	T <sub>0</sub> [s]	Priemer IoU [%]
Obr. 4.5(a)	3.01 7.26	76.10 63.08	Obr. 4.9(a)	----- -----	0.00 0.04
Obr. 4.5(b)	----- 7.50	8.04 47.11	Obr. 4.9(b)	37.33 12.43	45.75 55.03
Obr. 4.5(c)	0.033 22.60	64.03 34.76	Obr. 4.9(c)	0.00 2.76	65.30 74.88
Obr. 4.5(d)	14.35 5.70	49.86 62.57	Obr. 4.9(d)	----- 0.00	19.50 56.44

Tabuľka 4.1 Súhrn vyhodnotenia jednotlivých defektov

## 5. ZÁVER

V bakalárskej práci boli popísané rôzne metódy určené k detekcii defektov kamery jedného snímku prípadne sekvencie snímok. Všetky metódy spomínané v texte sú použiteľné na videozáznam akejkolvek kamery.

Bola vytvorená databáza 28 záznamov z motorového vozidla pri rôznych veľkostiach defektov priehľadného i nepriehľadného charakteru.

Priemerný čas  $T_0$  cez všetky záznamy funkcie *drops\_detect\_sobel* je rovný 12.05 s pričom priemerná hodnota IoU je rovná 26.06 % a priemerná maximálna hodnota IoU je rovná 36.56 %. Pri funkcii *drops\_detect\_difference* je  $T_0 = 22.78$  s, priemerná IoU = 42.94 % a priemerná maximálna hodnota IoU je rovná 58.54 %. Pri verifikovaných defektoch je  $T_0 = 17.14$  s, priemerná IoU = 32.70 % a priemerná maximálna hodnota IoU je rovná 44.88 %.

Na základe variability simulovaných defektov boli vytvorené metódy určené k detekcii a klasifikácii defektu. Tabuľka 4.1 zhrňa výsledky detekcie vybraných záznamov. V ľavej časti tabuľky sa nachádzajú defekty priehľadného charakteru. Z priemerných hodnôt výsledkov cez všetky záznamy je zrejmé, že funkcia *drops\_detect\_sobel* detekuje defekty rýchlejšie. Funkcia má lepšie výsledky pri detekcii priehľadných defektov. Funkcia *drops\_detect\_difference* má pomalší nábeh avšak výstup z detekcie obsahuje menej šumu.

Väčší problém po lokalizácii defektu bolo stanovenie jeho presnej veľkosti. Metódy nie sú schopné zaznamenať defekt podobnému defektu z obr. 4.9(a). Pri správnosti detekcie nezáleží ani tak na veľkosti defektu ale skôr na tom, aké veľké množstvo hmoty (vody / bahna) sa na sklíčku kamery nachádza. Je zložitá detekovať tenkú vrstvu defektu, ktorá nemá skoro žiadny vplyv na kvalitu obrazu.

Výstupom bakalárskej práce je algoritmus, ktorý detekuje, lokalizuje a klasifikuje defekty v dátach kamery spolu s databázou snímok simulácií rozličných typov defektov.

Datasetsy obrázkov boli vytvorené za pomerne prijateľných podmienkach počasia. Je otázne akým spôsobom bude defekt klasifikovaný v prípade dažďa, snehu, hmly prípadne iných podmienkach znižujúcich kvalitu záznamu nezávisle na defekte, ako napríklad jazda v nočných hodinách.



## REFERENCIE

- [1] S. You, S. Member, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi, "Adherent Raindrop Modeling , Detection and Removal in Video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1721–1733, 2016.
- [2] C. F. Chang, J. L. Wu, K. J. Chen, and M. C. Hsu, "A hybrid defect detection method for compact camera lens," *Adv. Mech. Eng.*, vol. 9, no. 8, pp. 1–19, 2017.
- [3] BABNIČ, Patrik. *Možnosti využití neuronových sítí v síťových prvcích*. Brno, 2011. Diplomová práce. VUT. Vedoucí práce Ing. Petr Vychodil.
- [4] F. Al Machot, M. Ali, A. Haj Mosa, C. Schwarzmüller, M. Gutmann, and K. Kyamakya, "Real-time raindrop detection based on cellular neural networks for ADAS," *J. Real-Time Image Process.*, pp. 1–13, 2016.
- [5] H. Kurihata *et al.*, "Rainy weather recognition from in-vehicle camera images for driver assistance," *IEEE Intell. Veh. Symp. Proc.*, vol. 2005, pp. 205–210, 2005.
- [6] RISHAV, Kumar. *Understanding Principal Component Analysis* [online]. Jan 2, 2018 [cit. 2019-01-08]. Dostupné z: <https://medium.com/@aptrishu/understanding-principle-component-analysis-e32be0253ef0>
- [7] RAY, Sunil. *Understanding Support Vector Machine algorithm from examples* [online]. September 13, 2017 [cit. 2019-01-09]. Dostupné z: <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>
- [8] *Understanding Cross-Correlation, Auto-Correlation, Normalization and Time Shift* [online]. March 8, 2016 [cit. 2019-01-09]. Dostupné z: <https://anomaly.io/understand-auto-cross-correlation-normalized-shift/>
- [9] Paul Bourke. *Cross Correlation* [online]. August 1996 [cit. 2019-01-09]. Dostupné z: <http://paulbourke.net/miscellaneous/correlate/>
- [10] J. L. Barron and N. a Thacker, "Tutorial: Computing 2D and 3D optical flow," *Imaging Sci. Biomed. Eng. Div. Med. Sch. Univ. Manchester*, no. 2004, pp. 1–12, 2005.
- [11] *Optical Flow* [online]. [cit. 2019-01-09]. Dostupné z: [https://docs.opencv.org/3.4/d7/d8b/tutorial\\_py\\_lucas\\_kanade.html](https://docs.opencv.org/3.4/d7/d8b/tutorial_py_lucas_kanade.html)

[12] ROSEBROCK, Adrian. *Intersection over Union (IoU) for object detection* [online]. November 7, 2016 [cit. 2019-05-16]. Dostupné z: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>